

PC-WiFi-01

サンプルプログラム解説

第1.1版 2017年09月04日

目次

1. 概要	1
1.1 概要.....	1
1.2 動作環境	2
1.3 ネットワーク構成イメージ図.....	3
1.4 動作モード.....	4
1.5 開発環境について.....	6
1.6 ワークスペースについて.....	6
2. サンプルプログラムの構成.....	7
2.1 フォルダ構成	7
2.2 ファイル構成	8
3. TCP/IP 通信サンプルプログラム.....	14
3.1 ビルド・デバッグ方法.....	14
3.2 動作説明 (TCP/IP 通信)	17
3.2.1 サンプルプログラム概要 (TCP/IP 通信 アドホックモード)	17
3.2.2 サンプルプログラム概要 (TCP/IP 通信 インフラストラクチャモード)	19
3.2.3 TCP/IP 通信エコーバックサーバ動作.....	20
3.3 RAM 動作時のメモリマップ (TCP/IP 通信サンプルプログラム共通)	22
3.4 ROM 動作時のメモリマップ (TCP/IP 通信サンプルプログラム共通)	23
4. UDP 通信サンプルプログラム	24
4.1 ビルド・デバッグ方法 (UDP 通信サンプルプログラム)	24
4.2 動作説明 (UDP 通信)	27
4.2.1 サンプルプログラム概要 (UDP 通信 アドホックモード)	27
4.2.2 サンプルプログラム概要 (UDP 通信 インフラストラクチャモード)	29
4.2.3 UDP 通信エコーバックサーバ動作	30
4.3 RAM 動作時のメモリマップ (UDP 通信サンプルプログラム共通)	32
4.4 ROM 動作時のメモリマップ (UDP 通信サンプルプログラム共通)	33

5. PC- WIFI -01 制御方法.....	34
5.1 概要.....	34
5.2 シリアルインタフェース.....	35
5.2.1 シリアルインタフェース仕様.....	35
5.2.2 シリアルインタフェースフロー.....	35
5.3 AT コマンド.....	36
5.4 オートボーレート.....	37
5.5 ファームウェアアップデート.....	37
5.6 コマンドリファレンス.....	38
5.6.1 Band コマンド.....	38
5.6.2 Init コマンド.....	38
5.6.3 Scan コマンド.....	39
5.6.4 Set Network Type コマンド.....	40
5.6.5 Pre Shared Key コマンド.....	40
5.6.6 Authentication Mode コマンド.....	41
5.6.7 Join コマンド.....	41
5.6.8 Set IP Parameters コマンド.....	43
5.6.9 Open a Listening TCP Server コマンド.....	44
5.6.10 Open a Listening UDP Socket コマンド.....	44
5.6.11 Send data to a Socket コマンド.....	45
5.6.12 Receive data on a Socket.....	46
5.6.13 Close a Socket コマンド.....	47
5.6.14 Disassociate コマンド.....	47
5.6.15 Soft Reset コマンド.....	47
5.6.16 Query Firmware Version コマンド.....	48
5.6.17 Power Mode コマンド.....	48

1. 概要

1.1 概要

本アプリケーションノートでは、PC-WiFi-01 に付属するサンプルプログラム Ver1.1 以降について解説します。

PC-WiFi-01 には、弊社製 AP-SH2A-0A 用サンプルプログラムが付属しています。

本サンプルプログラムの概要を以下に示します

サンプルプログラム	動作内容
TCP/IP 通信サンプルプログラム (アドホックモード クリエータ)	<ul style="list-style-type: none">・TCP/IP 通信エコーバックサーバ・シリアル通信・タイマ割り込み
TCP/IP 通信サンプルプログラム (アドホックモード ジョイナー)	<ul style="list-style-type: none">・TCP/IP 通信エコーバックサーバ・シリアル通信・タイマ割り込み
TCP/IP 通信サンプルプログラム (インフラストラクチャモード)	<ul style="list-style-type: none">・TCP/IP 通信エコーバックサーバ・シリアル通信・タイマ割り込み
UDP 通信サンプルプログラム (アドホックモード クリエータ)	<ul style="list-style-type: none">・UDP 通信エコーバックサーバ・シリアル通信・タイマ割り込み
UDP 通信サンプルプログラム (アドホックモード ジョイナー)	<ul style="list-style-type: none">・UDP 通信エコーバックサーバ・シリアル通信・タイマ割り込み
UDP 通信サンプルプログラム (インフラストラクチャモード)	<ul style="list-style-type: none">・UDP 通信エコーバックサーバ・シリアル通信・タイマ割り込み

1.2 動作環境

各サンプルプログラムの動作確認に必要な機器を以下に記します。

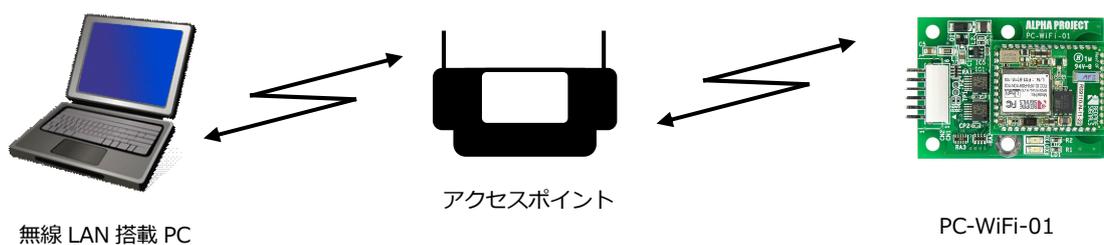
サンプルプログラム	動作確認に必要な機器
TCP/IP 通信サンプルプログラム (アドホックモード クリエータ)	・アドホック通信可能なホスト PC
TCP/IP 通信サンプルプログラム (アドホックモード ジョイナー)	・アドホック通信可能なホスト PC
TCP/IP 通信サンプルプログラム (インフラストラクチャモード)	・ネットワーク通信可能なホスト PC ・アクセスポイント
UDP 通信サンプルプログラム (アドホックモード クリエータ)	・アドホック通信可能なホスト PC
UDP 通信サンプルプログラム (アドホックモード ジョイナー)	・アドホック通信可能なホスト PC
UDP 通信サンプルプログラム (インフラストラクチャモード)	・ネットワーク通信可能なホスト PC ・アクセスポイント

1.3 ネットワーク構成イメージ図

以下に、「インフラストラクチャ」と「アドホック」時のネットワーク構成イメージ図を示します。

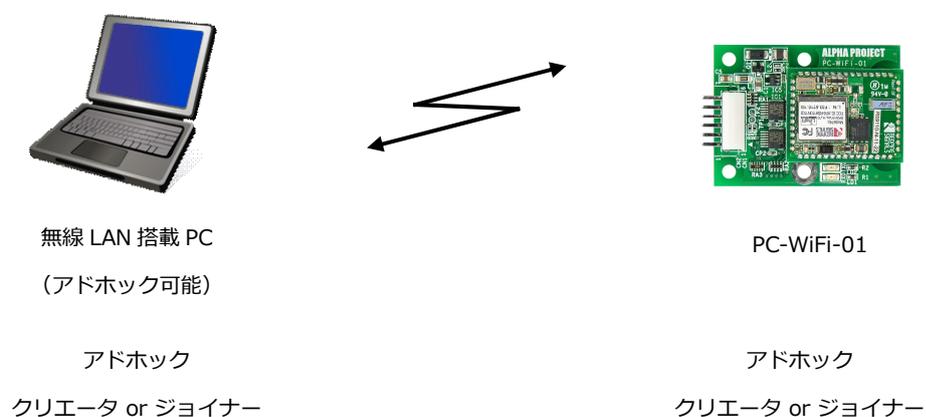
・インフラストラクチャ

アクセスポイント経由で、無線通信を行います。



・アドホック

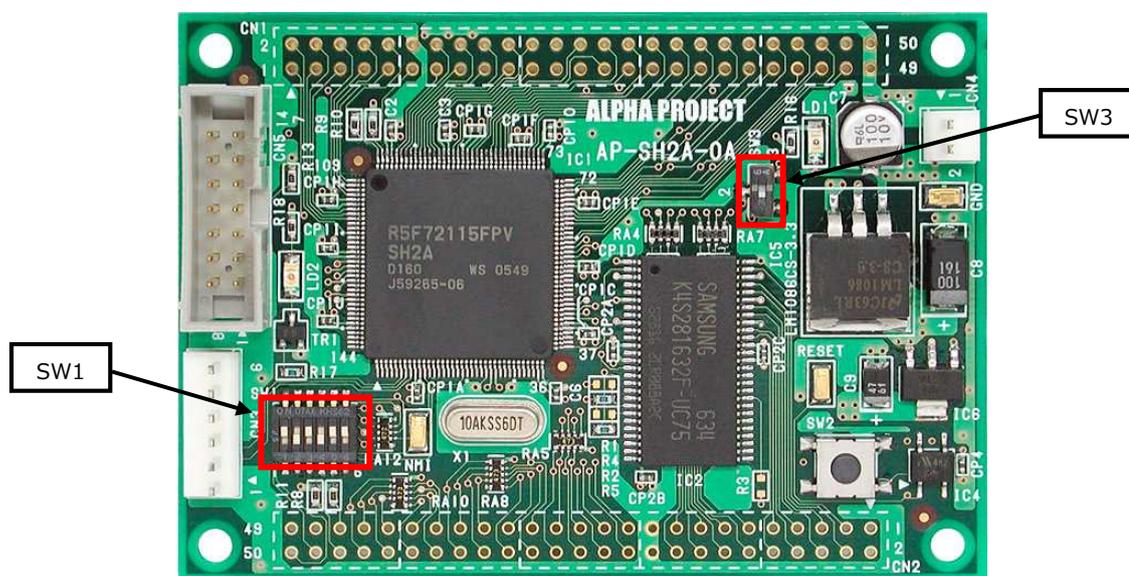
アドホック通信では、クリエータ（親）となった機器に、ジョイナー（子）となった機器が接続して通信を行います。



1.4 動作モード

本サンプルプログラムは、AP-SH2A-0A で動作します。CPU 動作モード、各メモリ設定は下記のようになっています。モードの設定方法等につきましては、「AP-SH2A-0A ハードウェアマニュアル」をご覧ください。
なお、下記以外の条件で動作させる場合には、ソースファイルやコンパイラオプションなどを変更する必要があります。

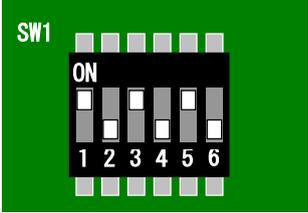
CPU 動作モード	: モード 2
デバッグモード	: OFF
クロックモード	: モード 6
外付け SDRAM 設定	: 外付け SDRAM を使用する



CPU ボードの設定を製品出荷時の状態とし、使用方法に合わせて以下の各スイッチの設定を行ってください。

プログラム動作時

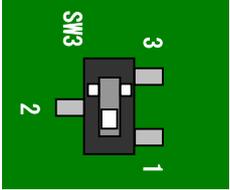
• SW1



<SW1 設定>

クロックモード	: モード6
CPU 動作モード	: モード2
デバッグモード	: OFF
FLASH 書き込み	: 禁止

• SW3

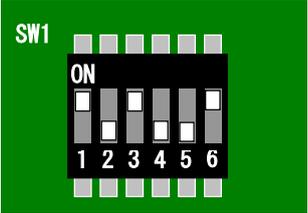


<SW3 設定>

ボード上の SDRAM	: 使用する
-------------	--------

プログラム書き込み時

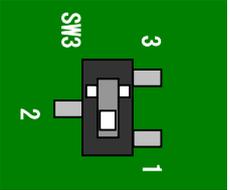
• SW1



<SW1 設定>

クロックモード	: モード6
CPU 動作モード	: モード2
デバッグモード	: ON
FLASH 書き込み	: 許可

• SW3



<SW3 設定>

ボード上の SDRAM	: 使用する
-------------	--------

Fig 1.4-1 動作モード設定

1.5 開発環境について

本サンプルプログラムは総合開発環境 High-performance Embedded Workshop（以下 HEW と表記します）を用いて開発されています。

サンプルプログラムに対応する開発環境、コンパイラのバージョンは次のようになります。

開発環境	バージョン	コンパイラ名	バージョン	備考
High-performance Embedded Workshop	V 3.01.08.000 以降	SHC ※1	V9.0.1.0 以降	SuperH ファミリ用 C/C++コンパイラパッケージに付属

※1：「SuperH ファミリ用 C/C++コンパイラパッケージ」です。ルネサスエレクトロニクス社のウェブサイトより評価版をダウンロードできます。

1.6 ワークスペースについて

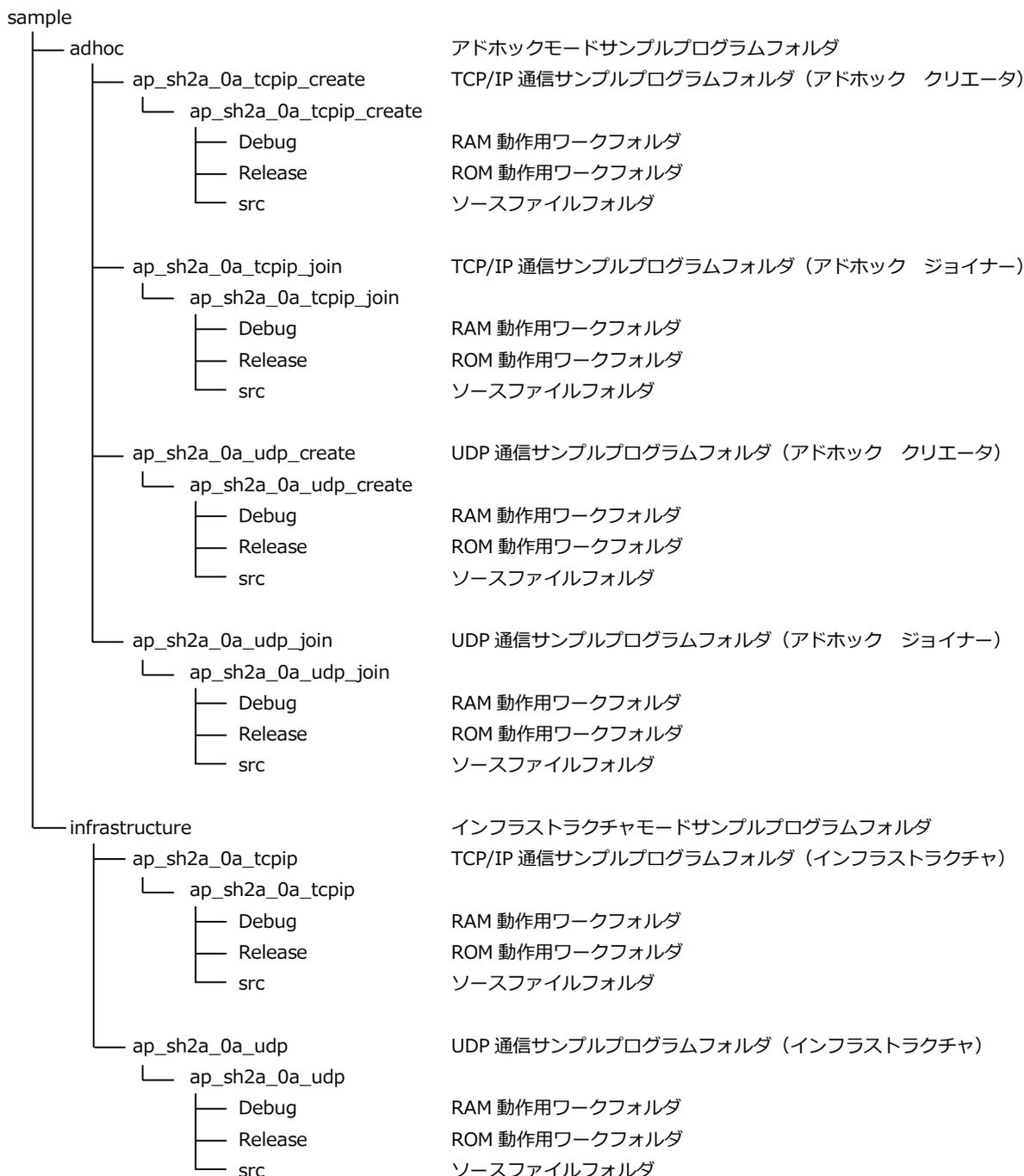
本サンプルプログラムの総合開発環境 High-performance Embedded Workshop ワークスペースは次のフォルダに格納されています。

サンプルプログラム	フォルダ
TCP/IP サンプルプログラム (アドホックモード クリエータ)	¥sample¥adhoc¥ap_sh2a_0a_tcpip_create¥ap_sh2a_0a_tcpip_create.hws
TCP/IP サンプルプログラム (アドホックモード ジョイナー)	¥sample¥adhoc¥ap_sh2a_0a_tcpip_join¥ap_sh2a_0a_tcpip_join.hws
UDP サンプルプログラム (アドホックモード クリエータ)	¥sample¥adhoc¥ap_sh2a_0a_udp_create¥ap_sh2a_0a_udp_create.hws
UDP サンプルプログラム (アドホックモード ジョイナー)	¥sample¥adhoc¥ap_sh2a_0a_udp_join¥ap_sh2a_0a_udp_join.hws
TCP/IP サンプルプログラム (インフラストラクチャモード)	¥sample¥infrastructure¥ap_sh2a_0a_tcpip¥ap_sh2a_0a_tcpip.hws
UDP サンプルプログラム (インフラストラクチャモード)	¥sample¥infrastructure¥ap_sh2a_0a_udp¥ap_sh2a_0a_udp.hws

2. サンプルプログラムの構成

2.1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。



2.2 ファイル構成

サンプルプログラムは以下のファイルで構成されています。

<¥sample フォルダ内>

adhoc	…	アドホックモードサンプルプログラム
infrastructure	…	インフラストラクチャモードサンプルプログラム
XcrossFinder_sh2a_0a.xfc	…	XcrossFinder 用コマンドファイル

<¥sample¥adhoc フォルダ内>

ap_sh2a_0a_tcpip_create	…	TCP/IP 通信サンプルプログラムフォルダ (アドホック クリエータ)
ap_sh2a_0a_tcpip_join	…	TCP/IP 通信サンプルプログラムフォルダ (アドホック ジョイナー)
ap_sh2a_0a_udp_create	…	UDP 通信サンプルプログラムフォルダ (アドホック クリエータ)
ap_sh2a_0a_udp_join	…	UDP 通信サンプルプログラムフォルダ (アドホック ジョイナー)

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_create フォルダ内>

ap_sh2a_0a_tcpip_create.hws	…	TCP/IP 通信サンプルプログラム HEW 用ワークスペース ファイル (アドホック クリエータ)
-----------------------------	---	---

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_create¥ap_sh2a_0a_tcpip_create フォルダ内>

ap_sh2a_0a_tcpip_create.hwp	…	TCP/IP 通信サンプルプログラム HEW 用プロジェクト ファイル (アドホック ジョイナー)
-----------------------------	---	--

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_create¥ap_sh2a_0a_tcpip_create¥Debug フォルダ内>

ap_sh2a_0a_tcpip_create.abs	…	TCP/IP 通信サンプルプログラム RAM 動作用オブジェクト ファイル (アドホック クリエータ) (elf 形式)
ap_sh2a_0a_tcpip_create.mot	…	TCP/IP 通信サンプルプログラム RAM 動作用 モトローラ S フォーマット形式ファイル (アドホック クリエータ)
ap_sh2a_0a_tcpip_create.map	…	TCP/IP 通信サンプルプログラム RAM 動作用マップファイ ル (アドホック クリエータ) コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_create¥ap_sh2a_0a_tcpip_create¥Release フォルダ内>

ap_sh2a_0a_tcpip_create.abs	…	TCP/IP 通信サンプルプログラム ROM 動作用オブジェクト ファイル (アドホック クリエータ) (elf 形式)
ap_sh2a_0a_tcpip_create.mot	…	TCP/IP 通信サンプルプログラム ROM 動作用 モトローラ S フォーマット形式ファイル (アドホック クリエータ)
ap_sh2a_0a_tcpip_create.map	…	TCP/IP 通信サンプルプログラム ROM 動作用マップファイ ル (アドホック クリエータ) コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_create¥ap_sh2a_0a_tcpip_create¥src フォルダ内>

7211.h	...	SH7211 内部レジスタ定義ヘッダファイル
boot.c	...	CPU 初期化処理ファイル
BoardDepend.h	...	ボード依存ファイル
common.h	...	共通ヘッダファイル
main.c	...	メイン処理ファイル
NetworkSetting.h	...	ネットワーク設定定義ファイル
sci.c	...	シリアル処理ファイル
section.src	...	セクション定義ファイル
timer.c	...	タイマ処理ファイル
vector.c	...	割込ベクタテーブルファイル
wifi.c	...	WiFi モジュールサンプルドライバファイル
wifi.h	...	WiFi モジュールサンプルドライバヘッダファイル

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_join フォルダ内>

ap_sh2a_0a_tcpip_join.hws	...	TCP/IP 通信サンプルプログラム HEW 用ワークスペース ファイル (アドホック ジョイナー)
---------------------------	-----	---

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_join¥ap_sh2a_0a_tcpip_join フォルダ内>

ap_sh2a_0a_tcpip_join.hwp	...	TCP/IP 通信サンプルプログラム HEW 用プロジェクト ファイル (アドホック ジョイナー)
---------------------------	-----	--

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_join¥ap_sh2a_0a_tcpip_join¥Debug フォルダ内>

ap_sh2a_0a_tcpip_join.abs	...	TCP/IP 通信サンプルプログラム RAM 動作用オブジェクト ファイル (アドホック ジョイナー) (elf 形式)
ap_sh2a_0a_tcpip_join.mot	...	TCP/IP 通信サンプルプログラム RAM 動作用 モトローラ S フォーマット形式ファイル (アドホック ジョイナー)
ap_sh2a_0a_tcpip_join.map	...	TCP/IP 通信サンプルプログラム RAM 動作用マップファイ ル (アドホック ジョイナー) コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_join¥ap_sh2a_0a_tcpip_join¥Release フォルダ内>

ap_sh2a_0a_tcpip_join.abs	...	TCP/IP 通信サンプルプログラム ROM 動作用オブジェクト ファイル (アドホック ジョイナー) (elf 形式)
ap_sh2a_0a_tcpip_join.mot	...	TCP/IP 通信サンプルプログラム ROM 動作用 モトローラ S フォーマット形式ファイル (アドホック ジョイナー)
ap_sh2a_0a_tcpip_join.map	...	TCP/IP 通信サンプルプログラム ROM 動作用マップファイ ル (アドホック ジョイナー) コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_tcpip_join¥ap_sh2a_0a_tcpip_join¥src フォルダ内>

7211.h	...	SH7211 内部レジスタ定義ヘッダファイル
boot.c	...	CPU 初期化処理ファイル
BoardDepend.h	...	ボード依存ファイル
common.h	...	共通ヘッダファイル
main.c	...	メイン処理ファイル
NetworkSetting.h	...	ネットワーク設定定義ファイル
sci.c	...	シリアル処理ファイル
section.src	...	セクション定義ファイル
timer.c	...	タイマ処理ファイル
vector.c	...	割込ベクタテーブルファイル
wifi.c	...	WiFi モジュールサンプルドライバファイル
wifi.h	...	WiFi モジュールサンプルドライバヘッダファイル

<¥sample¥adhoc¥ap_sh2a_0a_udp_create フォルダ内>

ap_sh2a_0a_udp_create.hws	...	UDP 通信サンプルプログラム HEW 用ワークスペースファイル (アドホック クリエータ)
---------------------------	-----	---

<¥sample¥adhoc¥ap_sh2a_0a_udp_create¥ap_sh2a_0a_udp_create フォルダ内>

ap_sh2a_0a_udp_create.hwp	...	UDP 通信サンプルプログラム HEW 用プロジェクトファイル (アドホック クリエータ)
---------------------------	-----	--

<¥sample¥adhoc¥ap_sh2a_0a_udp_create¥ap_sh2a_0a_udp_create¥Debug フォルダ内>

ap_sh2a_0a_udp_create.abs	...	UDP 通信サンプルプログラム RAM 動作用オブジェクト ファイル (アドホック クリエータ) (elf 形式)
ap_sh2a_0a_udp_create.mot	...	UDP 通信サンプルプログラム RAM 動作用 モトローラ S フォーマット形式ファイル (アドホック クリエータ)
ap_sh2a_0a_udp_create.map	...	UDP 通信サンプルプログラム RAM 動作用マップファイル (アドホック クリエータ) コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_udp_create¥ap_sh2a_0a_udp_create¥Release フォルダ内>

ap_sh2a_0a_udp_create.abs	...	UDP 通信サンプルプログラム ROM 動作用オブジェクト ファイル (アドホック クリエータ) (elf 形式)
ap_sh2a_0a_udp_create.mot	...	UDP 通信サンプルプログラム ROM 動作用 モトローラ S フォーマット形式ファイル (アドホック クリエータ)
ap_sh2a_0a_udp_create.map	...	UDP 通信サンプルプログラム ROM 動作用マップファイル (アドホック クリエータ) コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_udp_create¥ap_sh2a_0a_udp_create¥src フォルダ内>

7211.h	...	SH7211 内部レジスタ定義ヘッダファイル
boot.c	...	CPU 初期化処理ファイル
BoardDepend.h	...	ボード依存ファイル
common.h	...	共通ヘッダファイル
main.c	...	メイン処理ファイル
NetworkSetting.h	...	ネットワーク設定定義ファイル
sci.c	...	シリアル処理ファイル
section.src	...	セクション定義ファイル
timer.c	...	タイマ処理ファイル
vector.c	...	割込ベクタテーブルファイル
wifi.c	...	WiFi モジュールサンプルドライバ
wifi.h	...	WiFi モジュールサンプルドライバヘッダファイル

<¥sample¥adhoc¥ap_sh2a_0a_udp_join フォルダ内>

ap_sh2a_0a_udp_join.hws	...	UDP 通信サンプルプログラム HEW 用ワークスペースファイル (アドホック ジョイナー)
-------------------------	-----	---

<¥sample¥adhoc¥ap_sh2a_0a_udp_join¥ap_sh2a_0a_udp_join フォルダ内>

ap_sh2a_0a_udp_join.hwp	...	UDP 通信サンプルプログラム HEW 用プロジェクトファイル (アドホック ジョイナー)
-------------------------	-----	--

<¥sample¥adhoc¥ap_sh2a_0a_udp_join¥ap_sh2a_0a_udp_join¥Debug フォルダ内>

ap_sh2a_0a_udp_join.abs	...	UDP 通信サンプルプログラム RAM 動作オブジェクト ファイル (アドホック ジョイナー) (elf 形式)
ap_sh2a_0a_udp_join.mot	...	UDP 通信サンプルプログラム RAM 動作 モトローラ S フォーマット形式ファイル (アドホック ジョイナー)
ap_sh2a_0a_udp_join.map	...	UDP 通信サンプルプログラム RAM 動作マップファイル (アドホック ジョイナー) コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_udp_join ¥ap_sh2a_0a_udp_join¥Release フォルダ内>

ap_sh2a_0a_udp_join.abs	...	UDP 通信サンプルプログラム ROM 動作オブジェクト ファイル (アドホック ジョイナー) (elf 形式)
ap_sh2a_0a_udp_join.mot	...	UDP 通信サンプルプログラム ROM 動作 モトローラ S フォーマット形式ファイル (アドホック ジョイナー)
ap_sh2a_0a_udp_join.map	...	UDP 通信サンプルプログラム ROM 動作マップファイル (アドホック ジョイナー) コンパイル後は、.obj,.lib 等のファイルが生成されます

<¥sample¥adhoc¥ap_sh2a_0a_udp_join¥ap_sh2a_0a_udp_join¥src フォルダ内>

7211.h	...	SH7211 内部レジスタ定義ヘッダファイル
boot.c	...	CPU 初期化処理ファイル
BoardDepend.h	...	ボード依存ファイル
common.h	...	共通ヘッダファイル
main.c	...	メイン処理ファイル
NetworkSetting.h	...	ネットワーク設定定義ファイル
sci.c	...	シリアル処理ファイル
section.src	...	セクション定義ファイル
timer.c	...	タイマ処理ファイル
vector.c	...	割込ベクタテーブルファイル
wifi.c	...	WiFi モジュールサンプルドライバファイル
wifi.h	...	WiFi モジュールサンプルドライバヘッダファイル

<¥sample¥infrastructure フォルダ内>

ap_sh2a_0a_tcpipserver	...	TCP/IP 通信サンプルプログラムフォルダ
ap_sh2a_0a_udp	...	UDP 通信サンプルプログラムフォルダ

<¥sample¥infrastructure¥ap_sh2a_0a_tcpip フォルダ内>

ap_sh2a_0a_tcpip.hws	...	TCP/IP 通信サンプルプログラム HEW 用 ワークスペースファイル
----------------------	-----	---

<¥sample¥infrastructure¥ap_sh2a_0a_tcpipserver¥ap_sh2a_0a_tcpipserver フォルダ内>

ap_sh2a_0a_tcpip.hwp	...	TCP/IP 通信サンプルプログラム HEW 用 プロジェクトファイル
----------------------	-----	--

<¥sample¥infrastructure¥ap_sh2a_0a_tcpip¥ap_sh2a_0a_tcpip¥Debug フォルダ内>

ap_sh2a_0a_tcpip.abs	...	TCP/IP サンプルプログラム通信 RAM 動作用オブジェクト ファイル(elf 形式)
ap_sh2a_0a_tcpip.mot	...	TCP/IP 通信サンプルプログラム RAM 動作用 モトローラ S フォーマット形式ファイル
ap_sh2a_0a_tcpip.map	...	TCP/IP 通信サンプルプログラム RAM 動作用マップファイ ル コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥infrastructure¥ap_sh2a_0a_tcpip¥ap_sh2a_0a_tcpip¥Release フォルダ内>

ap_sh2a_0a_tcpip.abs	...	TCP/IP 通信サンプルプログラム ROM 動作用オブジェクト ファイル (elf 形式)
ap_sh2a_0a_tcpip.mot	...	TCP/IP 通信サンプルプログラム ROM 動作用 モトローラ S フォーマット形式ファイル
ap_sh2a_0a_tcpip.map	...	TCP/IP 通信サンプルプログラム ROM 動作用マップファイ ル コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥infrastructure¥ap_sh2a_0a_tcpip¥ap_sh2a_0a_tcpip¥src フォルダ内>

7211.h	...	SH7211 内部レジスタ定義ヘッダファイル
boot.c	...	CPU 初期化処理ファイル
BoardDepend.h	...	ボード依存ファイル
common.h	...	共通ヘッダファイル
main.c	...	メイン処理ファイル
NetworkSetting.h	...	ネットワーク設定定義ファイル
sci.c	...	シリアル処理ファイル
section.src	...	セクション定義ファイル
timer.c	...	タイマ処理ファイル
vector.c	...	割込ベクタテーブルファイル
wifi.c	...	WiFi モジュールサンプルドライバファイル
wifi.h	...	WiFi モジュールサンプルドライバヘッダファイル

<¥sample¥infrastructure¥ap_sh2a_0a_udp フォルダ内>

ap_sh2a_0a_udp.hws	...	UDP 通信サンプルプログラム HEW 用ワークスペースファイル
--------------------	-----	----------------------------------

<¥sample¥infrastructure¥ap_sh2a_0a_udp¥ap_sh2a_0a_udp フォルダ内>

ap_sh2a_0a_udp.hwp	...	UDP 通信サンプルプログラム HEW 用プロジェクトファイル
--------------------	-----	---------------------------------

<¥sample¥infrastructure¥ap_sh2a_0a_udp¥ap_sh2a_0a_udp¥Debug フォルダ内>

ap_sh2a_0a_udp.abs	...	UDP 通信サンプルプログラム RAM 動作オブジェクトファイル(elf 形式)
ap_sh2a_0a_udp.mot	...	UDP 通信サンプルプログラム RAM 動作モトローラ S フォーマット形式ファイル
ap_sh2a_0a_udp.map	...	UDP 通信サンプルプログラム RAM 動作マップファイル コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥infrastructure¥ap_sh2a_0a_udp¥ap_sh2a_0a_udp¥Release フォルダ内>

ap_sh2a_0a_udp.abs	...	UDP 通信サンプルプログラム ROM 動作オブジェクトファイル (elf 形式)
ap_sh2a_0a_udp.mot	...	UDP 通信サンプルプログラム ROM 動作モトローラ S フォーマット形式ファイル
ap_sh2a_0a_udp.map	...	UDP 通信サンプルプログラム ROM 動作マップファイル コンパイル後は、.obj, .lib 等のファイルが生成されます

<¥sample¥infrastructure¥ap_sh2a_0a_udp¥ap_sh2a_0a_udp¥src フォルダ内>

7211.h	...	SH7211 内部レジスタ定義ヘッダファイル
boot.c	...	CPU 初期化処理ファイル
BoardDepend.h	...	ボード依存ファイル
common.h	...	共通ヘッダファイル
main.c	...	メイン処理ファイル
NetworkSetting.h	...	ネットワーク設定定義ファイル
sci.c	...	シリアル処理ファイル
section.src	...	セクション定義ファイル
timer.c	...	タイマ処理ファイル
vector.c	...	割込ベクタテーブルファイル
wifi.c	...	WiFi モジュールサンプルドライバファイル
wifi.h	...	WiFi モジュールサンプルドライバヘッダファイル

3. TCP/IP 通信サンプルプログラム

3.1 ビルド・デバッグ方法

TCP/IP 通信サンプルプログラムのビルド・デバッグ方法を以下に記します。

アドホックモード（クリエイター・ジョイナー）、インフラストラクチャモードを問わず、ビルド・デバッグの方法は同一です。

(1) ビルド

- ① HEW を起動し、「Table 3.1-1 TCP/IP 通信サンプルプログラム HWS 一覧」を参考に、ビルド・デバッグを行うサンプルプログラムの HWS ファイルを読み込みます。

サンプルプログラムの種類	読み込むファイル
アドホックモード クリエータ	¥sample¥adhoc¥ap_sh2a_0a_tcpip_create¥ap_sh2a_0a_tcpip_create.hws
アドホックモード ジョイナー	¥sample¥adhoc¥ap_sh2a_0a_tcpip_join¥ap_sh2a_0a_tcpip_join.hws
インフラストラクチャモード	¥sample¥infrastructure¥ap_sh2a_0a_tcpip¥ap_sh2a_0a_tcpip.hws

Table 3.1-1 TCP/IP 通信サンプルプログラム HWS ファイル一覧

- ② 最初の読み込みを行ったときに、「ワークスペース（Workspace）が移動しました」という内容の確認メッセージが表示されますので「はい」を選択してください。
- ③ 最初の読み込みを行ったときに、コンパイラバージョンによって、バージョンの選択を行うダイアログが表示されることがあります。表示された場合には、使用するコンパイラバージョンを選択してください。
- ④ [ビルド]ボタン横のリストボックス[Configuration Section]から、[Debug]または[Release]を選択します。
[Debug]を選択した場合、¥Debug ワークフォルダ内に RAM 動作のオブジェクトが生成されます。
[Release]を選択した場合、¥Release ワークフォルダ内に ROM 動作のオブジェクトが生成されます。
- ⑤ メニューの [ビルド] - [ビルド] を実行してください。モトローラファイル（拡張子が mot のファイル）、アブソリュートファイル（拡張子が abs のファイル）が出力されます。
このとき、マップファイルはワークフォルダに作成されます。

HEW の詳細な使用方法につきましては、HEW のマニュアルを参照してください。

(2) RAM 上でのデバッグ

- ① AP-SH2A-0A のスイッチを、「Fig 1.3-1 動作モード設定」の「プログラム書き込み時」の状態に設定します。
- ② XsSight を起動し、¥sample フォルダ直下にある XrossFinder_sh2a_0a.xfc コマンドファイルを読み込みます。
- ③ 「(1) ビルド」で出力したワークスペースの ¥Debug フォルダ内のデバッグを行うアブソリュートファイルを XsSight からダウンロードして動作を確認してください。

(3) ROM 上でのデバッグ

- ① AP-SH2A-0A のスイッチを、「Fig 1.3-1 動作モード設定」の「プログラム書き込み時」の状態に設定します。
¥sample フォルダ内の XrossFinder_sh2a_0a.xfc と「(1) ビルド」で出力したワークスペースの ¥Release フォルダ内のアブソリュートファイルを XsSight で読み込みます。
- ② XsSight のメニューから FlashWriterEX を選択し、下図 Fig3.1-1 のように設定を行ってください。
- ③ START ボタンを押してプログラムの書き込みを行い、動作を確認してください。

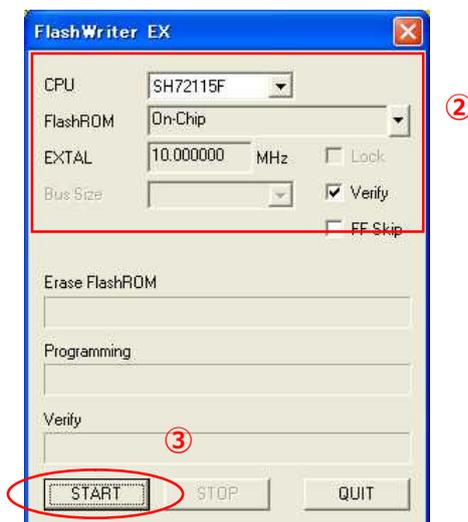


Fig3.1-1 FlashWriterEX for XsSight の設定

(4) XsSight 未使用時の確認方法

・FlashWriterEX を使用する場合

- ① アダプタ (HJ-LINK / XrossFinder / XrossFinder Evo) を使用して PC とボードを繋ぎます。
- ② AP-SH2A-0A のスイッチを、「Fig1.4-1 動作モード設定」の「プログラム書き込み時」の状態に設定します。
- ③ FlashWriterEX を起動して、「Table3.1-2 FlashWriterEX の設定」を参考に設定を行ってください。
- ④ FlashWriterEX で、¥sample フォルダ直下にある XrossFinder_sh2a_0a.xfc コマンドファイルを使用するように設定してください。
- ⑤ ¥Release フォルダ内のモトローラファイルをボードに書き込みます。
- ⑥ AP-SH2A-0A のスイッチを「Fig1.4-1 動作モード設定」の「プログラム動作時」の状態に設定し、動作を確認してください。

※ FlashWriterEX の使用方法の詳細につきましては、FlashWriterEX のマニュアルを参照してください。

アダプタ設定	XrossFinder Evo 使用時は「XrossFinder Evo」 XrossFinder 使用時は「XrossFinder」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinder 使用時のみ)	5MHz 以下
CPU	SH72115F
EXTAL	10.0MHz
FLASHROM	On-Chip

Table3.1-2 FlashWriterEX の設定

3.2 動作説明 (TCP/IP 通信)

サンプルプログラムの動作を以下にフローチャート図で記します。

図中に記載されている動作やコマンドについては「5. PC-WiFi-01 制御方法」を参照してください。

3.2.1 サンプルプログラム概要 (TCP/IP 通信 アドホックモード)

TCP/IP 通信サンプルプログラム (アドホックモード) は、下記の動作を行います。

1) クリエータ

- SCIF1 に接続された PC-WiFi-01 に対してコマンドを送信し、TCP/IP エコーバックサーバを構築します。その後、PC-WiFi-01 が受信したデータをそのまま送信元へ送信します。
動作確認は、ホスト PC 上のターミナルソフト (Tera Term など) を使用して行ってください。
※ TCP/IP エコーバックサーバ動作の詳細は、「3.2.3 TCP/IP 通信エコーバックサーバ動作」を参照してください。

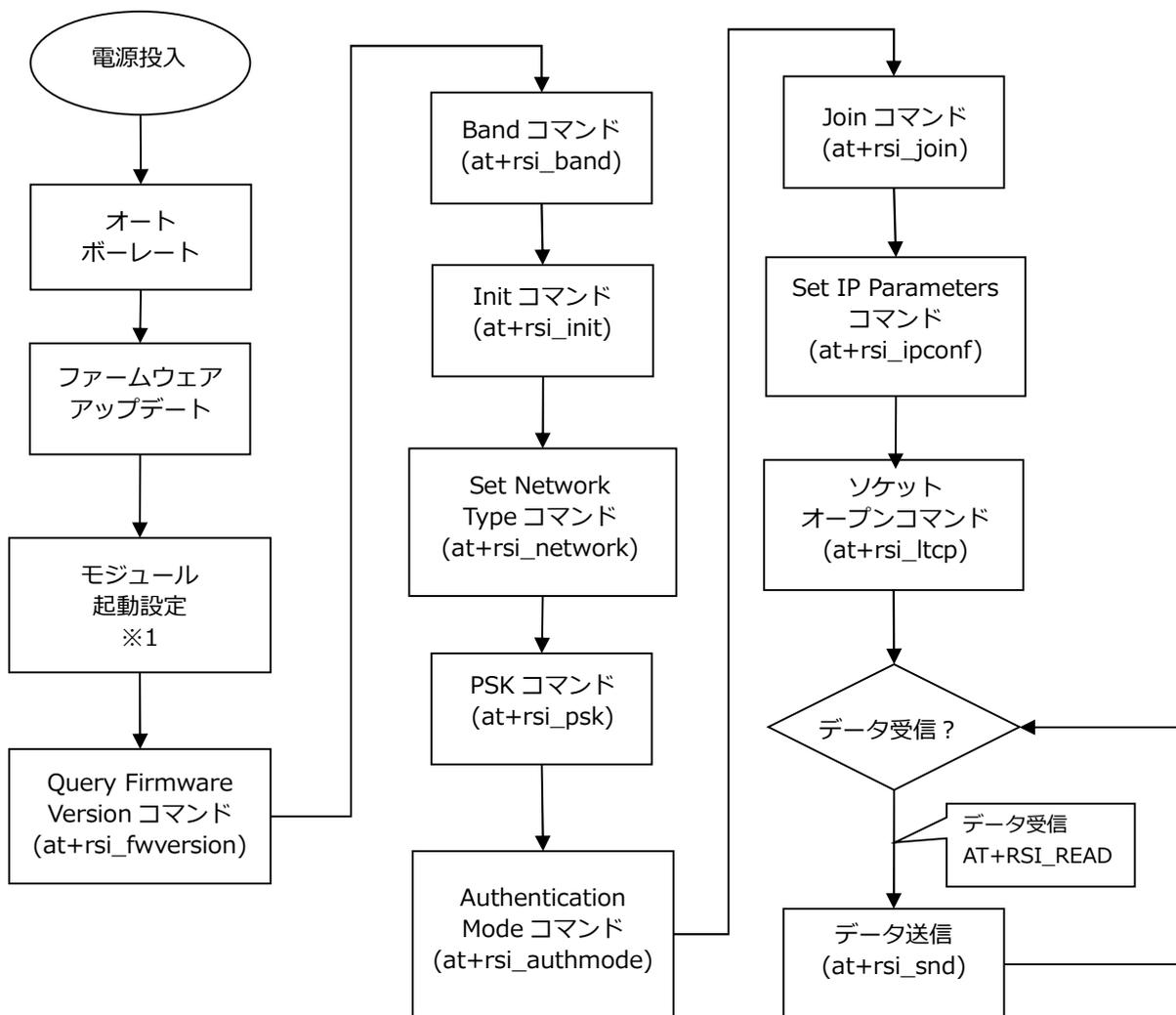


Fig 3.2-1 TCP/IP 通信サンプルプログラム アドホッククリエイータ PC-WiFi-01 制御フロー

2) ジョイナー ※ジョイナー動作時は、クリエータ動作時と比べて「scan」処理が加わります。

- SCIF1 に接続された PC-WiFi-01 に対してコマンドを送信し、アドホックモードでアクセスポイントに接続した後、TCP/IP エコーバックサーバを構築します。
その後、PC-WiFi-01 が受信したデータをそのまま送信元に送信します。
動作確認は、ホスト PC 上のターミナルソフト（Tera Term など）を使用して行ってください。
※ TCP/IP エコーバックサーバ動作の詳細は、「3.2.3 TCP/IP 通信エコーバックサーバ動作」を参照してください。

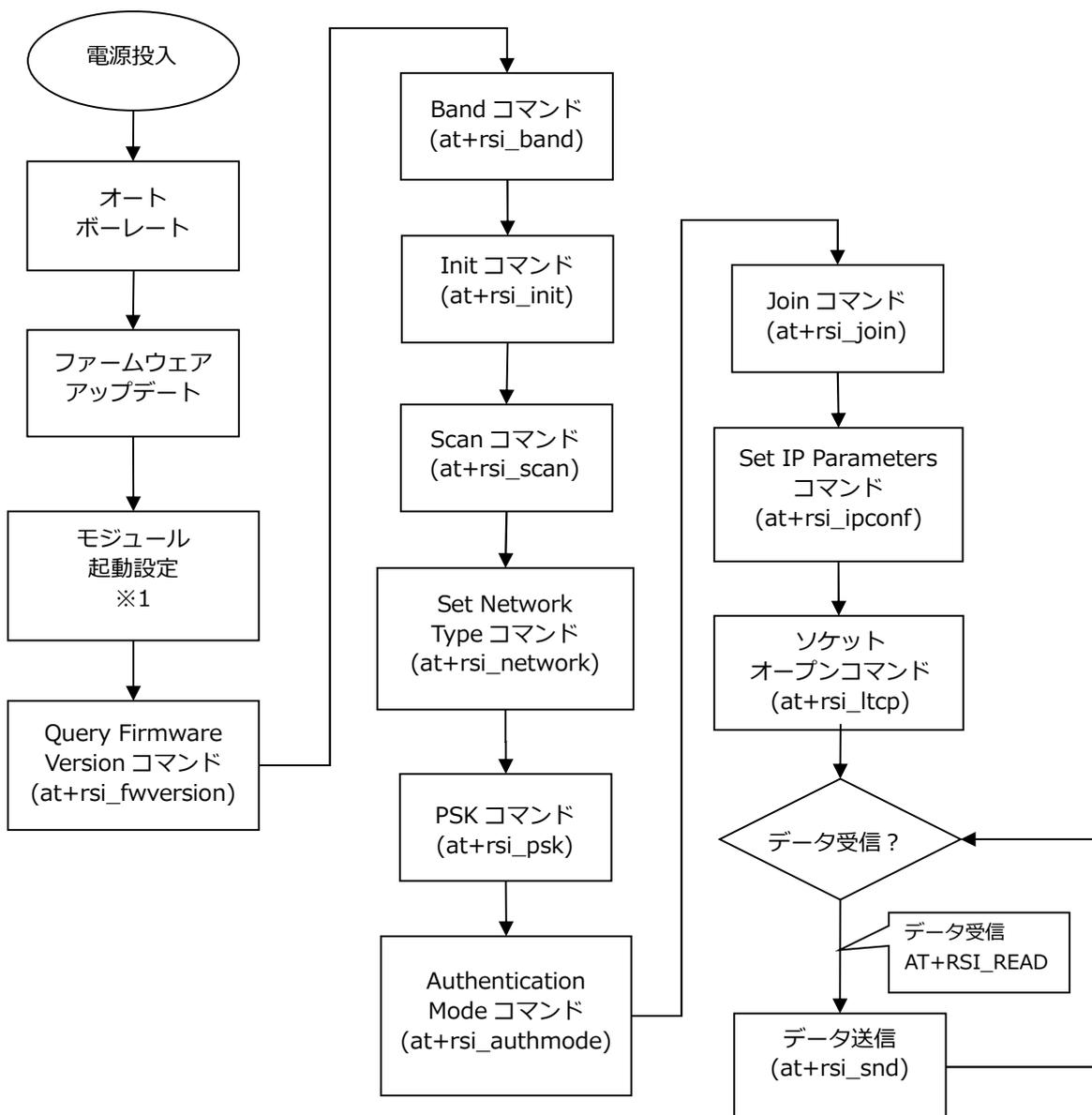


Fig 3.2-2 TCP/IP 通信サンプルプログラム アドホックジョイナー PC-WiFi-01 制御フロー

3.2.2 サンプルプログラム概要 (TCP/IP 通信 インフラストラクチャモード)

TCP/IP 通信サンプルプログラム (インフラストラクチャモード) は、下記の動作を行います。

- SCIF1 に接続された PC-WiFi-01 に対してコマンドを送信し、インフラストラクチャモードでアクセスポイントに接続した後、TCP/IP エコーバックサーバを構築します。その後、受信したデータをそのまま送信元に送信します。
動作確認は、ホスト PC 上のターミナルソフト (Tera Term など) を使用して行ってください。
※ TCP/IP エコーバックサーバ動作の詳細は、「3.2.3 TCP/IP 通信エコーバックサーバ動作」を参照してください。

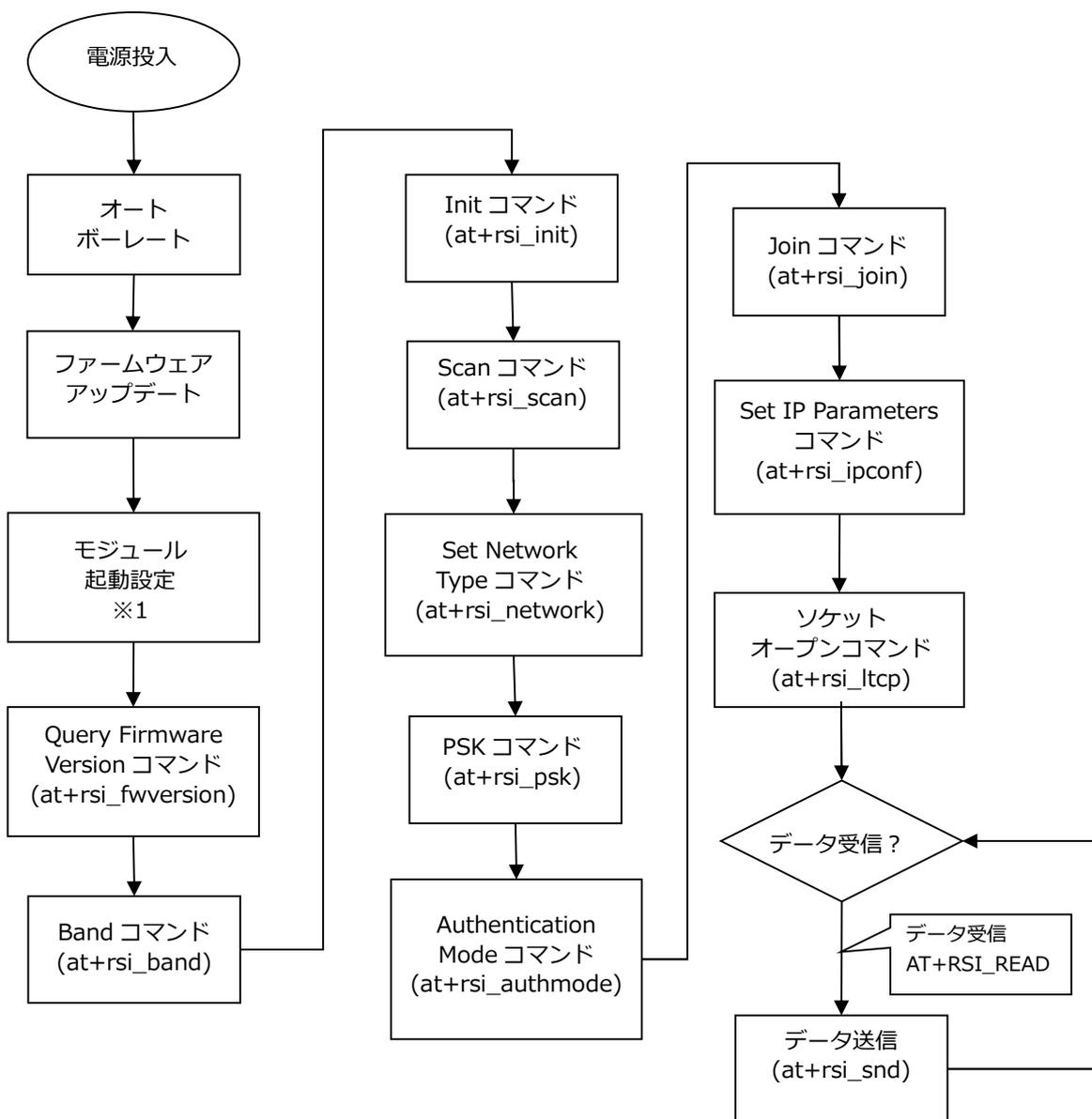


Fig 3.2-3 TCP/IP 通信サンプルプログラム インフラストラクチャ PC-WiFi-01 制御フロー

※1 モジュール起動設定は、下記のコマンドを実行しています。モジュール起動時に必ず行ってください。

- | | |
|--|--|
| ① Band コマンド(at+rsi_band) | ② Init コマンド(at+rsi_init) |
| ③ Set NetworkType コマンド(at+rsi_network) | ④ Power Mode コマンド(at+rsi_mode) |
| ⑤ Scan コマンド(at+rsi_scan) | ⑥ Reset コマンド(at+rsi_reset) |

3.2.3 TCP/IP 通信工コーバックサーバ動作

(1) TCP/IP ネットワーク設定

以下にサンプルプログラムのネットワーク設定を記します。※

TCP/IP ネットワーク設定 (アドホックモード)	
使用帯域	2.4GHz
使用チャンネル	3ch
ネットワーク接続	アドホックモード
送信レート	自動設定
送信レベル	ミディアムレベル
PSK	PC-WiFi-01SamplePSK
アクセスポイント	PC-WiFi-01Sample
IP アドレス	192.168.1.200
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.1.253
PC-WiFi-01 でオープンする ソケットのポート番号	8000

Table3.2-1 TCP/IP 通信ネットワーク設定 (アドホックモード)

TCP/IP ネットワーク設定 (インフラストラクチャモード)	
使用帯域	2.4GHz
使用チャンネル	3ch
ネットワーク接続	インフラストラクチャモード
送信レート	自動設定
送信レベル	ミディアムレベル
PSK	PC-WiFi-01SamplePSK
アクセスポイント	PC-WiFi-01Sample
IP アドレス	192.168.1.200
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.1.253
PC-WiFi-01 でオープンする ソケットのポート番号	8000

Table3.2-2 TCP/IP 通信ネットワーク設定 (インフラストラクチャモード)

- ※ これらの設定は弊社の環境において設定した値となっています。ご利用時は、お使いの環境のネットワーク管理者に問い合わせ、ご利用の環境に沿ってそれぞれ適切な値を設定しビルドしてください。
- ※ これらの設定値は、各サンプルプログラムのソースフォルダ内にある「NetworkSetting.h」にて定義されています。

(2) TCP/IP 通信エコーバックサーバ動作 (アドホックモード クリエータ)

以下の手順に従い、TCP/IP 通信エコーバックサーバの動作を確認してください。

- ① CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ② プログラムがソケットオープンコマンドまで完了すると、CPU ボードの LD2 が点灯します。
CPU ボードの LD2 が点灯したことを確認し、アドホック通信機器の設定を行います。
その際、使用する設定は「**Table 3.2-1 TCP/IP 通信ネットワーク設定 (アドホックモード)**」で設定した値となります。
- ③ アドホック機器を無線 LAN ネットワークに接続し、エコーバックが行われることを確認してください。
- ④ 以上で TCP/IP 通信エコーバックサーバ動作 (アドホックモード クリエータ) の動作は終了です。

(3) TCP/IP 通信エコーバックサーバ動作 (アドホックモード ジョイナー)

以下の手順に従い、TCP/IP 通信エコーバックサーバの動作を確認してください。

- ① アドホック通信機器の設定を行い、無線 LAN ネットワークに接続します。
その際、使用する設定は「**Table 3.2-1 TCP/IP 通信ネットワーク設定 (アドホックモード)**」で設定した値となります。
- ② CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ③ プログラムがソケットオープンコマンドまで完了すると、CPU ボードの LD2 が点灯します。
CPU ボードの LD2 が点灯したことを確認し、アドホック通信機器の側からエコーバックが行われることを確認してください。
- ④ 以上で TCP/IP 通信エコーバックサーバ動作 (アドホックモード ジョイナー) の動作は終了です。

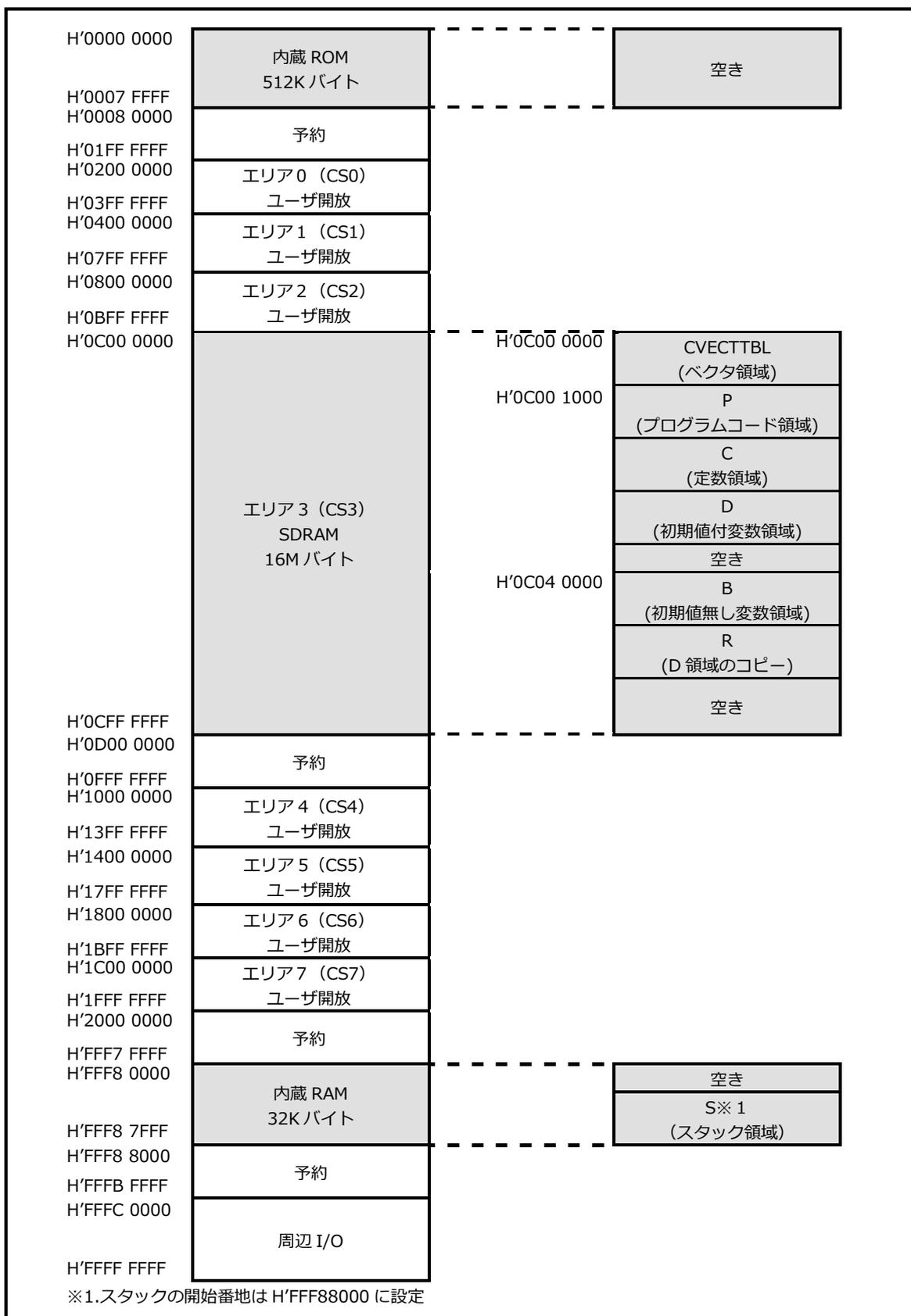
(4) TCP/IP 通信エコーバックサーバ動作 (インフラストラクチャモード)

以下の手順に従い、TCP/IP 通信エコーバックサーバの動作を確認してください。

- ① CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ② ホスト PC 上でターミナルソフト (Tera Term など) を起動し、Ethernet 接続の設定を行います。
その際、使用する設定は「**Table 3.2-2 TCP/IP 通信ネットワーク設定 (インフラストラクチャモード)**」で設定した値となります。
- ③ プログラムがソケットオープンコマンドまで完了すると、CPU ボードの LD2 が点灯します。
CPU ボードの LD2 が点灯したことを確認し、ターミナルソフトを使用して、エコーバックが行われることを確認してください。
- ④ 以上で TCP/IP 通信エコーバックサーバ動作 (インフラストラクチャモード) の動作は終了です。

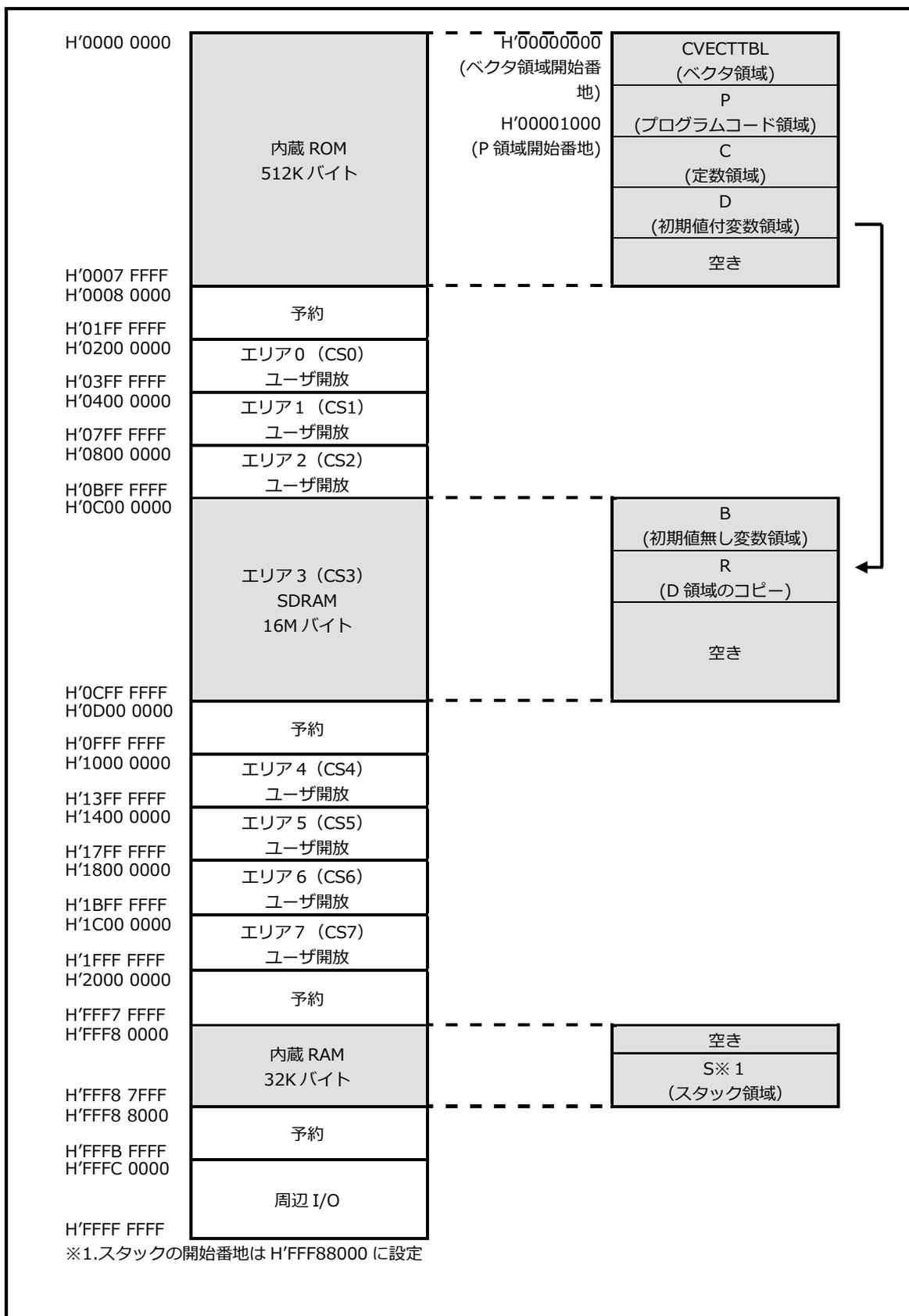
3.3 RAM 動作時のメモリマップ (TCP/IP 通信サンプルプログラム共通)

メモリマップを以下に示します。



3.4 ROM 動作時のメモリマップ (TCP/IP 通信サンプルプログラム共通)

メモリマップを以下に示します。



4. UDP 通信サンプルプログラム

4.1 ビルド・デバッグ方法（UDP 通信サンプルプログラム）

UDP 通信サンプルプログラムのビルド・デバッグ方法を以下に記します。

アドホックモード、インフラストラクチャモードを問わず、ビルド・デバッグの方法は同一です。

（1）ビルド

- ① HEW を起動し、「Table 4.1-1 UDP 通信サンプルプログラム HWS ファイル一覧」を参考に、ビルド・デバッグを行うサンプルプログラムの HWS ファイルを読み込みます。

サンプルプログラムの種類	読み込むファイル
アドホックモード クリエータ	¥sample¥adhoc¥ap_sh2a_0a_udp_create¥ap_sh2a_0a_udp_create.hws
アドホックモード ジョイナー	¥sample¥adhoc¥ap_sh2a_0a_udp_join¥ap_sh2a_0a_udp_join.hws
インフラストラクチャモード	¥sample¥infrastructure¥ap_sh2a_0a_udp¥ap_sh2a_0a_udp.hws

Table 4.1-1 UDP 通信サンプルプログラム HWS ファイル一覧

- ② 最初の読み込みを行ったときに、「ワークスペース（Workspace）が移動しました」という内容の確認メッセージが表示されますので「はい」を選択してください。
- ③ 最初の読み込みを行ったときに、コンパイラバージョンによって、バージョンの選択を行うダイアログが表示されることがあります。表示された場合には、使用するコンパイラバージョンを選択してください。
- ④ [ビルド]ボタン横のリストボックス[Configuration Section]から、[Debug]または[Release]を選択します。
[Debug]を選択した場合、¥Debug ワークフォルダ内に RAM 動作のオブジェクトが生成されます。
[Release]を選択した場合、¥Release ワークフォルダ内に ROM 動作のオブジェクトが生成されます。
- ⑤ メニューの [ビルド] - [ビルド] を実行してください。モトローラファイル（拡張子が mot のファイル）、アブソリュートファイル（拡張子が abs のファイル）が出力されます。このとき、マップファイルはワークフォルダに作成されます。

HEW の詳細な使用方法につきましては、HEW のマニュアルを参照してください。

（2）RAM 上でのデバッグ

- ① AP-SH2A-0A のスイッチを、「Fig 1.3-1 動作モード設定」の「プログラム書き込み時」の状態に設定します。
- ② XsSight を起動し、¥sample フォルダ直下にある XrossFinder_sh2a_0a.xfc コマンドファイルを読み込みます。
- ③ 「（1）ビルド」で出力したアブソリュートファイルを XsSight からダウンロードして動作を確認してください。

(3) ROM 上でのデバッグ

- ① AP-SH2A-0A のスイッチを、「Fig 1.3-1 動作モード設定」の「プログラム書き込み時」の状態に設定します。
- ② ¥sample フォルダ内の XrossFinder_sh2a_0a.xfc と「(1) ビルド」で出力したワークスペースの¥Release フォルダ内のアブソリュートファイルを XsSight で読み込みます。
- ③ XsSight のメニューから FlashWriterEX を選択し、下図 Fig4.1-1 のように設定を行ってください。
- ④ START ボタンを押してプログラムの書き込みを行い、動作を確認してください。

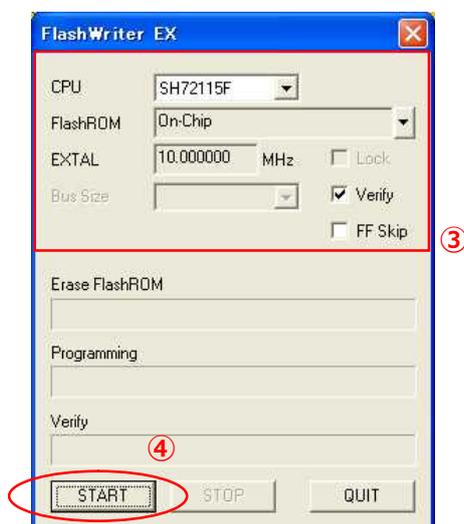


Fig4.1-1 FlashWriterEX for XsSight の設定

(4) XsSight 未使用時の確認方法

・ FlashWriterEX を使用する場合

- ① アダプタ (HJ-LINK / XrossFinder / XrossFinder Evo) を使用して PC とボードを繋ぎます。
- ② AP-SH2A-0A のスイッチを、「Fig 1.4-1 動作モード設定」の「プログラム書き込み時」の状態に設定します。
- ③ FlashWriterEX を起動して、「Table4.1-2 FlashWriterEX の設定」を参考に設定を行ってください。
- ④ FlashWriterEX で、¥sample フォルダ直下にある XrossFinder_sh2a_0a.xfc コマンドファイルを使用するように設定してください。
- ⑤ ¥Release フォルダ内のモトローラファイルをボードに書き込みます。
- ⑥ AP-SH2A-0A のスイッチを、「Fig 1.4-1 動作モード設定」の「プログラム動作時」の状態に設定し、動作を確認してください。

※ FlashWriterEX の使用方法の詳細につきましては、FlashWriterEX のマニュアルを参照してください。

アダプタ設定	XrossFinder Evo 使用時は「XrossFinder Evo」 XrossFinder 使用時は「XrossFinder」 HJ-LINK 使用時は「HJ-LINK」
JTAG クロック (XrossFinder 使用時のみ)	5MHz 以下
CPU	SH72115F
EXTAL	10.0MHz
FLASHROM	On-Chip

Table4.1-2 FlashWriterEX の設定

4.2 動作説明 (UDP 通信)

4.2.1 サンプルプログラム概要 (UDP 通信 アドホックモード)

UDP 通信サンプルプログラム (アドホックモード) は、下記の動作を行います。

1) クリエータ

- SCIF1 に接続された PC-WiFi-01 に対してコマンドを送信し、UDP ポートを開放します。
その後、UDP 通信で受信したデータをそのまま UDP 通信で送信元に送信します。
動作確認は、ホスト PC 上のターミナルソフト (SocketDebugger など) を使用して行ってください。
※ UDP エコーバックサーバ動作の詳細は、「4.2.3 UDP 通信エコーバックサーバ動作」を参照してください。

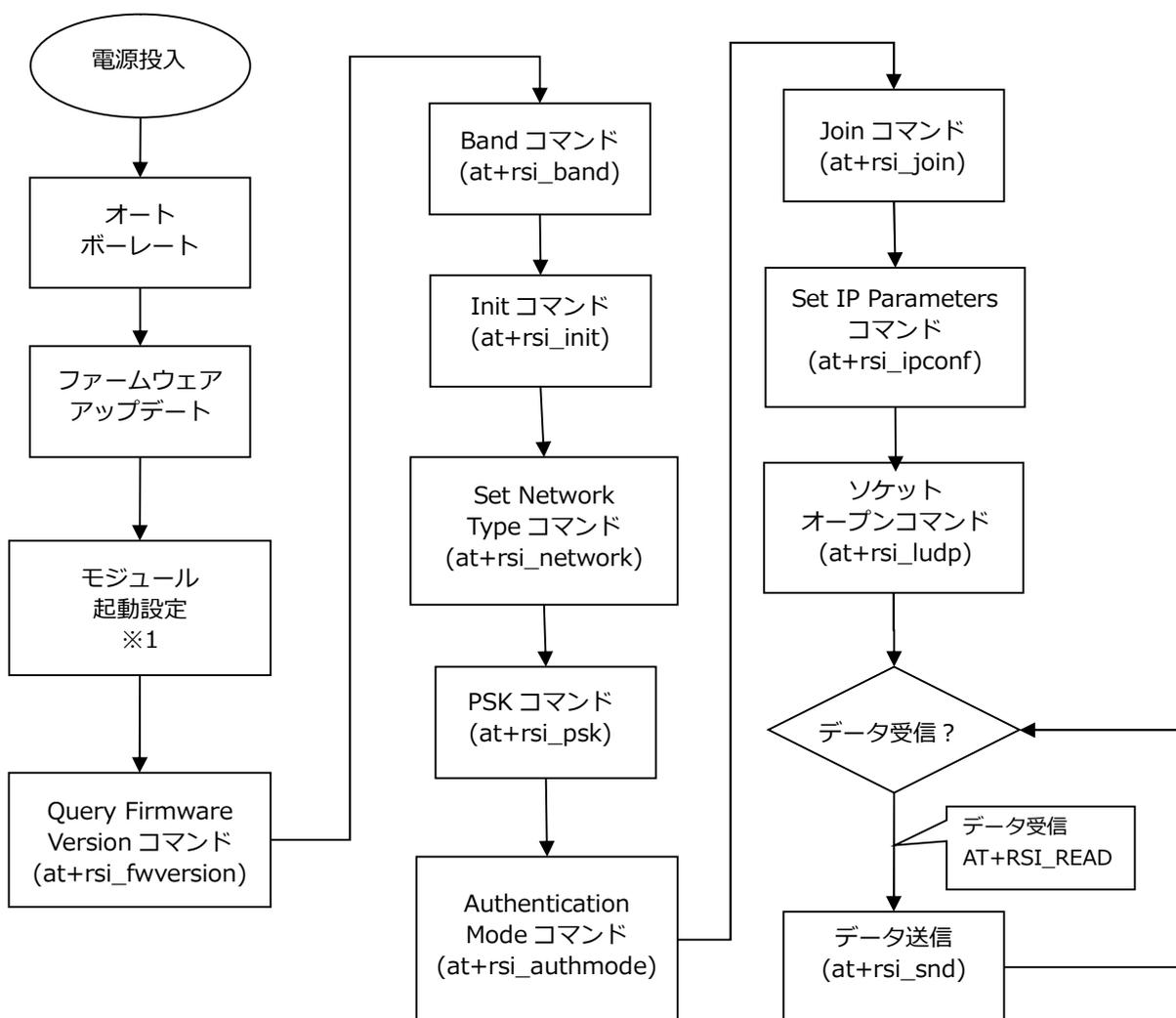


Fig 4.2-1 UDP 通信サンプルプログラム アドホッククリエイト PC-WiFi-01 制御フロー

2) ジョイナー

- SCIF1 に接続された PC-WiFi-01 に対してコマンドを送信し、アドホックモードでアクセスポイントに接続した後、UDP ポートを開放します。
その後、UDP 通信で受信したデータをそのまま UDP 通信で送信元に送信します。
動作確認は、ホスト PC 上のターミナルソフト (SocketDebugger など) を使用して行ってください。
※ UDP エコーバックサーバ動作の詳細は、「4.2.3 UDP 通信エコーバックサーバ動作」を参照してください。

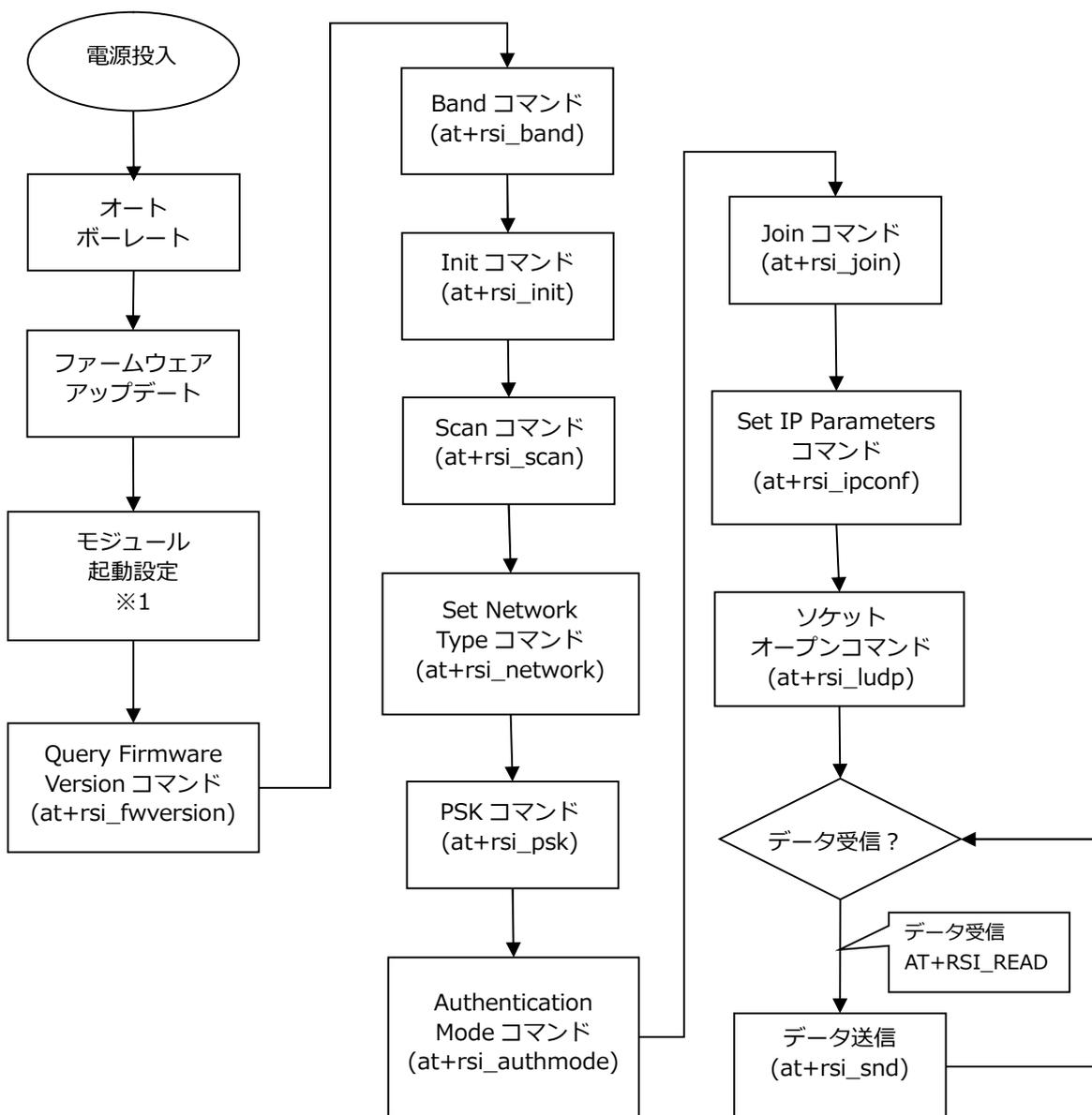


Fig 4.2-2 UDP 通信サンプルプログラム アドホックジョイン PC-WiFi-01 制御フロー

4.2.2 サンプルプログラム概要 (UDP 通信 インフラストラクチャモード)

TCP/IP 通信サンプルプログラム (インフラストラクチャモード) は、下記の動作を行います。

- SCIF1 に接続された PC-WiFi-01 に対してコマンドを送信し、インフラストラクチャモードでアクセスポイントに接続した後、UDP ポートを開放します。
その後、UDP 通信で受信したデータをそのまま UDP 通信で送信元に送信します。
動作確認は、ホスト PC 上のターミナルソフト (SocketDebugger など) を使用して行ってください。
※ UDP エコーバックサーバ動作の詳細は、「4.2.3 UDP 通信エコーバックサーバ動作」を参照してください。

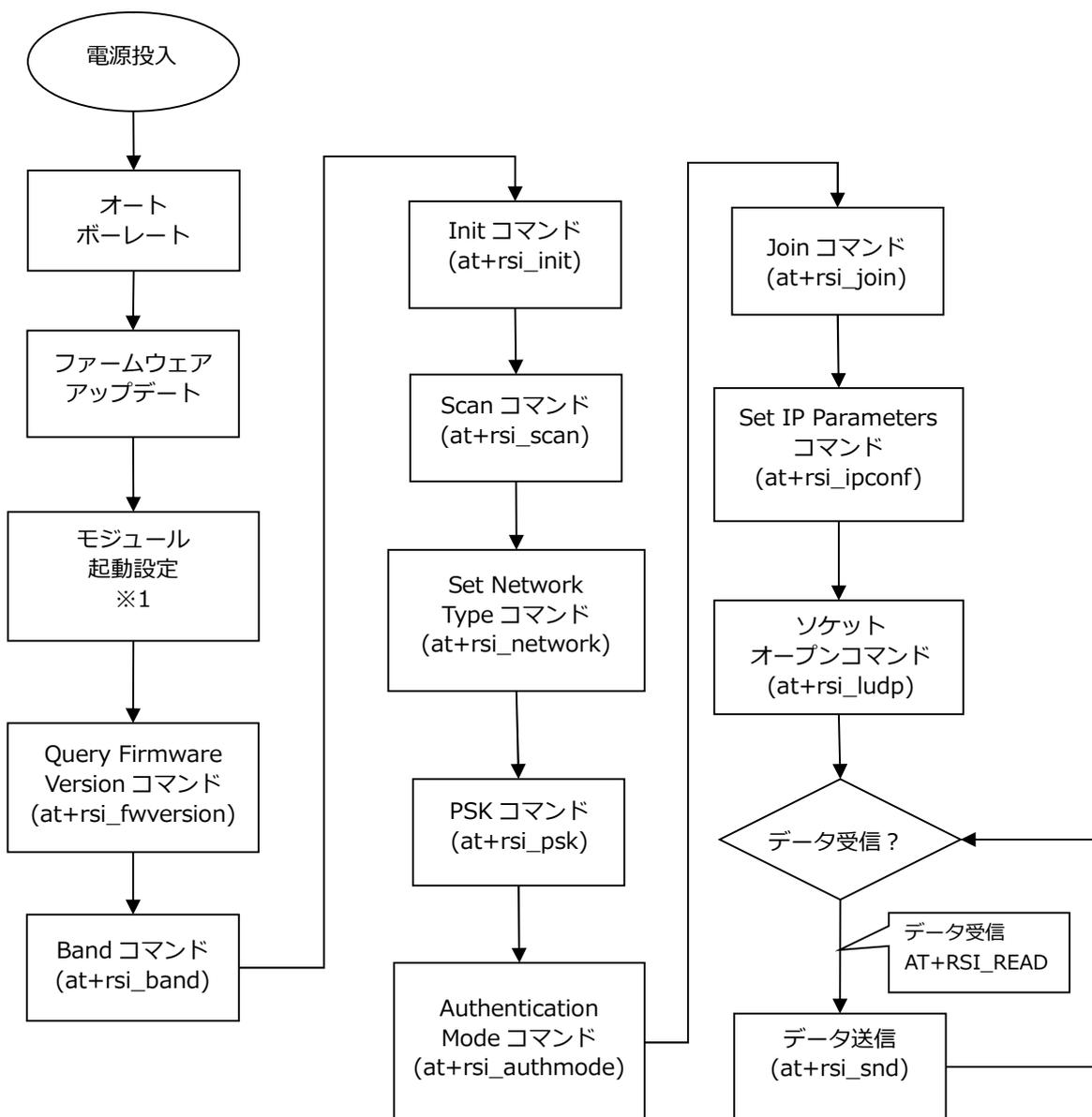


Fig 4.2-3 UDP 通信サンプルプログラム インフラストラクチャ PC-WiFi-01 制御フロー

※1 モジュール起動設定は、下記のコマンドを実行しています。モジュール起動時には必ず行ってください。

- ① [Band コマンド\(at+rsi_band\)](#)
- ② [Init コマンド\(at+rsi_init\)](#)
- ③ [Set NetworkType コマンド\(at+rsi_network\)](#)
- ④ [Power Mode コマンド\(at+rsi_mode\)](#)
- ⑤ [Scan コマンド\(at+rsi_scan\)](#)
- ⑥ [Reset コマンド\(at+rsi_reset\)](#)

4.2.3 UDP 通信工コーバックサーバ動作

(1) ネットワーク設定

以下にサンプルプログラムのネットワーク設定を記します。※

UDP 通信サンプルプログラムネットワーク設定 (アドホックモード)	
使用帯域	2.4GHz
使用チャンネル	3ch
ネットワーク接続	アドホックモード
送信レート	自動設定
送信レベル	ミディアムレベル
PSK	PC-WiFi-01SamplePSK
アクセスポイント	PC-WiFi-01Sample
IP アドレス	192.168.1.200
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.1.253
PC-WiFi-01 でオープンする ソケットのポート番号	8000

Table4.2-1 UDP 通信ネットワーク設定 (アドホックモード)

UDP 通信サンプルプログラムネットワーク設定 (インフラストラクチャモード)	
使用帯域	2.4GHz
使用チャンネル	3ch
ネットワーク接続	インフラストラクチャモード
送信レート	自動設定
送信レベル	ミディアムレベル
PSK	PC-WiFi-01SamplePSK
アクセスポイント	PC-WiFi-01Sample
IP アドレス	192.168.1.200
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.1.253
PC-WiFi-01 でオープンする ソケットのポート番号	8000

Table4.2-2 UDP 通信ネットワーク設定 (インフラストラクチャモード)

- ※ これらの設定は弊社環境において設定した値となっています。ご利用時は、お使いの環境のネットワーク管理者に問い合わせ、ご利用の環境に沿ってそれぞれ適切な値を設定してください。
- ※ これらの設定値は、各サンプルプログラムのソースフォルダ内にある「NetworkSetting.h」にて定義されています。

(2) UDP 通信エコーバックサーバ動作 (アドホックモード クリエータ)

以下の手順に従い、UDP 通信エコーバックサーバの動作を確認してください。

- ① CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ② プログラムがソケットオープンコマンドまで完了すると、CPU ボードの LD2 が点灯します。
CPU ボードの LD2 が点灯したことを確認し、アドホック通信機器の設定を行います。
その際、使用する設定は「Table 4.2-1 UDP 通信ネットワーク設定 (アドホックモード)」で設定した値となります。
- ③ アドホック機器を無線 LAN ネットワークに接続し、エコーバックが行われることを確認してください。
- ④ 以上で UDP 通信エコーバックサーバ動作 (アドホックモード クリエータ) の動作は終了です。

(3) UDP 通信エコーバックサーバ動作 (アドホックモード ジョイナー)

以下の手順に従い、UDP 通信エコーバックサーバの動作を確認してください。

- ① アドホック通信機器の設定を行い、無線 LAN ネットワークに接続します。
その際、使用する設定は「Table 4.2-1 UDP 通信ネットワーク設定 (アドホックモード)」で設定した値となります。
- ② CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ③ プログラムがソケットオープンコマンドまで完了すると、CPU ボードの LD2 が点灯します。
CPU ボードの LD2 が点灯したことを確認し、アドホック通信機器の側からエコーバックが行われることを確認してください。
- ④ 以上で UDP 通信エコーバックサーバ動作 (アドホックモード ジョイナー) の動作は終了です。

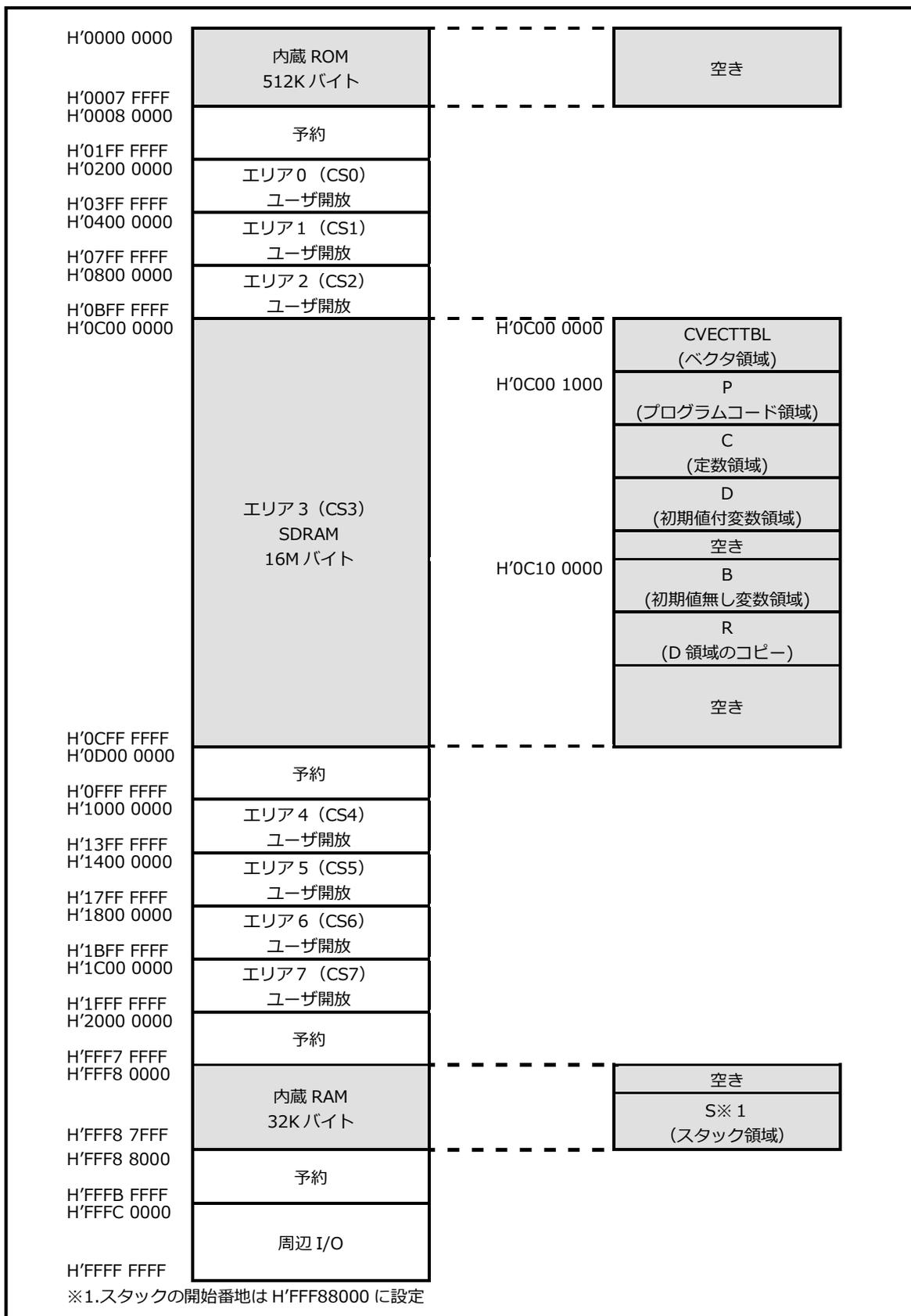
(4) UDP 通信エコーバックサーバ動作 (インフラストラクチャモード)

以下の手順に従い、UDP 通信エコーバックサーバの動作を確認してください。

- ① CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ② ホスト PC 上でターミナルソフト (Tera Term など) を起動し、Ethernet 接続の設定を行います。
その際、使用する設定は「Table 4.2-2 UDP 通信ネットワーク設定 (インフラストラクチャモード)」で設定した値となります。
- ③ プログラムがソケットオープンコマンドまで完了すると、CPU ボードの LD2 が点灯します。
CPU ボードの LD2 が点灯したことを確認し、ターミナルソフトを使用して、エコーバックが行われることを確認してください。
- ④ 以上で UDP 通信エコーバックサーバ動作 (インフラストラクチャモード) の動作は終了です。

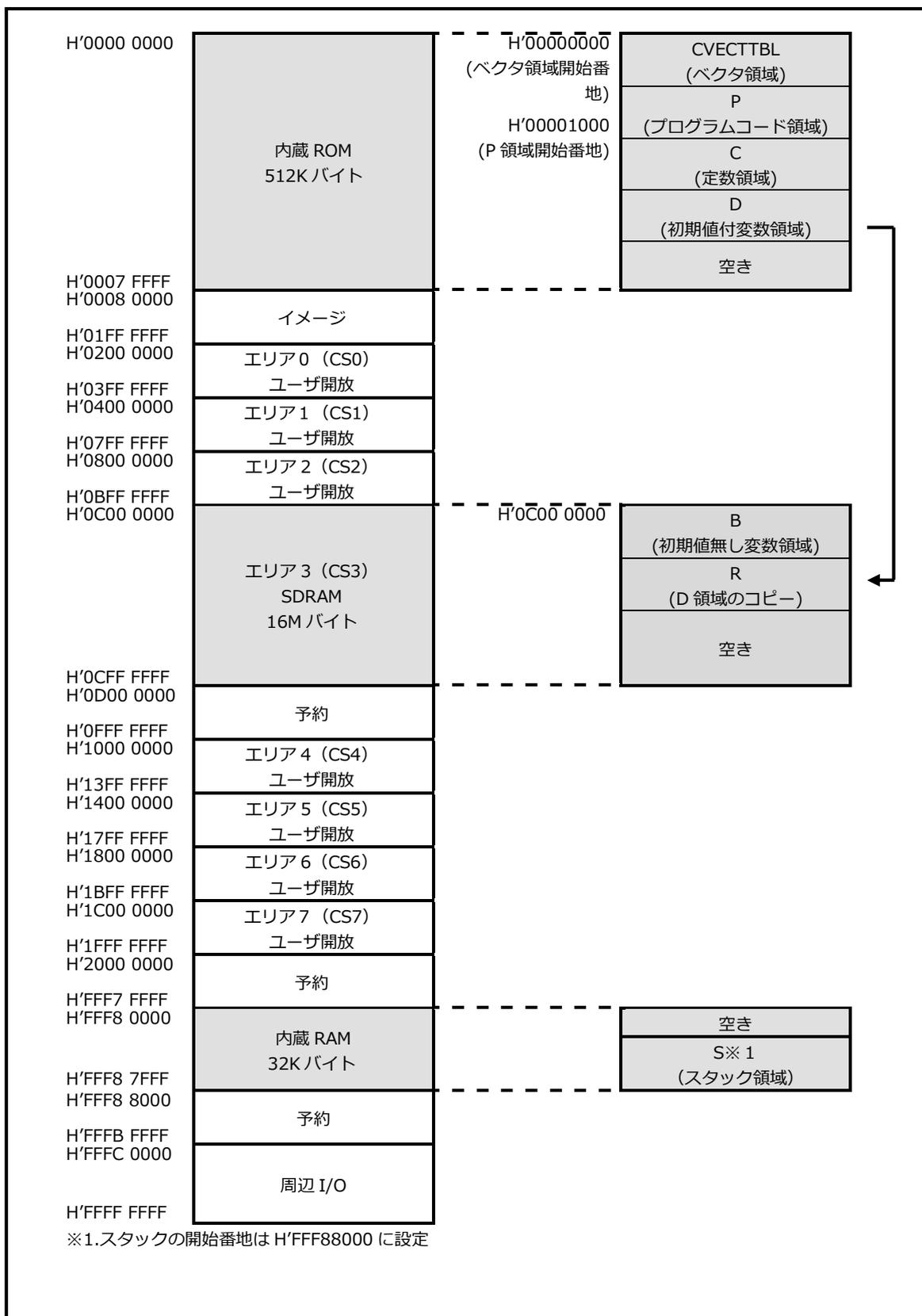
4.3 RAM 動作時のメモリマップ (UDP 通信サンプルプログラム共通)

メモリマップを以下に示します。



4.4 ROM 動作時のメモリマップ (UDP 通信サンプルプログラム共通)

メモリマップを以下に示します。



5. PC- WIFI -01 制御方法

5.1 概要

PC-WiFi-01 はホスト CPU とのインタフェースにシリアルインタフェースを採用しています。
ホスト CPU はシリアルインタフェースから ASCII コマンドを送信することで PC-WiFi-01 の操作を行い、初期化、ネットワークの設定、データの送受信などを行います。

5.2 シリアルインタフェース

5.2.1 シリアルインタフェース仕様

PC-WiFi-01 のシリアルインタフェース仕様を以下に記します。

機能	仕様
通信方式	調歩同期式
ボーレート	9600/19200/38400/57600/115200/200000/ 230400/460800/921600/1843200/ 3686400 bps
データビット	8 ビット (固定)
ストップビット	2 ビット (固定)
パリティ	なし (固定)
フロー制御	なし (固定)

Table 5.2-1 シリアルインタフェース仕様

5.2.2 シリアルインタフェースフロー

PC-WiFi-01 とホスト CPU とのシリアル通信は以下の流れで行われます。

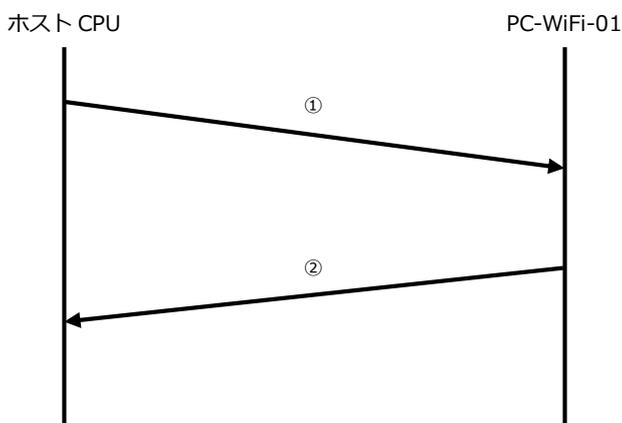


Fig 5.2-1 シリアルインタフェース制御フロー

- ① PC-WiFi-01 へのコマンド入力です。ASCII 文字列によるコマンドを送信することで、ホスト CPU から PC-WiFi-01 を制御することができます。
- ② PC-WiFi-01 からのレスポンスです。ASCII 文字列と Hex データの組み合わせでホスト CPU に応答が返されます。

※ 送信するコマンドおよびレスポンスの詳細に関しましては、「5.6 コマンドリファレンス」か、「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」を参照してください。

5.3 AT コマンド

PC-WiFi-01 が認識するコマンド（AT コマンド）は、ASCII の文字列によって構成されます。

以下にコマンド例を記します。

コマンドの詳細に関しては、「5.6 コマンドリファレンス」か、

「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」を参照してください。

at+rsi_band=0<0x0D><0x0A>

コマンド	説明
at+rsi_band=0	使用帯域を設定するコマンドです。例は 0 (=2.4GHz 帯) を使用する設定となります。 データは ASCII の文字列となります。
<0x0D><0x0A>	コマンドの最後に送信する CR(0x0D)と LF(0x0A)コードです。 この 2 つのコードを送信することで、PC-WiFi-01 はコマンドを認識します。

Table 5.3-1 シリアルコマンド例

コマンドに対して正常終了時、PC-WiFi-01 は以下のようなレスポンスを返します*。

OK<0x0D><0x0A>

コマンド	説明
OK	コマンドが正常に実行されたことを示す ASCII の文字列です。
<0x0D><0x0A>	レスポンスの最後に付属する CR(0x0D)と LF(0x0A)コードです。

Table 5.3-2 シリアルコマンドレスポンス例

* レスポンスの内容は先に送信したコマンドによって異なります。詳細に関しては、「5.6 コマンドリファレンス」か「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」を参照してください。

5.4 オートボーレート

オートボーレートは、PC-WiFi-01 の電源 ON 後に自動的に行われる処理です。

ホスト CPU と特定の通信を行うことにより、通信に用いるボーレートを自動で設定することが出来ます。

ホスト CPU は、「Table 5.2-1 シリアルインタフェース仕様」の仕様に合わせて通信パラメータを設定してください。

※ オートボーレート処理の詳細は、「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」の「2.1 Auto Baud Rate Detection (ABRD)」を参照してください。

- ① ホスト CPU と PC-WiFi-01 を接続し、電源を入れます。
- ② PC-WiFi-01 の起動が完了するまで 100ms 以上待機します。
- ③ ホスト CPU から<0x1C>を送信します。
- ④ ホスト CPU が<0x55>を受信することを確認します。
 - ※ ホスト CPU が<0x55>を受信出来ない場合、200ms 以上経過した後に再度 <0x1C>を送信します。
- ⑤ ホスト CPU から<0x55>を送信します。

以上でオートボーレートの設定は終了です。

※ 電源 ON 後、ホスト CPU から<0x1C>を送信しない場合、約 18 秒後に自動的にオートボーレート処理がタイムアウトとなり、ボーレートの値は 115200bps に設定されます。

5.5 ファームウェアアップデート

オートボーレートが終了した後、PC-WiFi-01 はファームウェアアップデートの確認を行います。

PC-WiFi-01 は以下のデータを送信しますので、<0x6E><0x0A>を送信するか、何もせずに待機してください。

```
<0x20>WELCOME<0x20>TO<0x20>REDPINE<0x20>SIGNALS<0x0D><0x0A>  
<0x20><0x0D><0x0A>  
<0x20>Firmware<0x20>upgrade<0x20>(y/n)<0x0C><0x0D><0x0A>
```

<0x6E><0x0A>を送信するか、何もせずに待機していると、PC-WiFi-01 は以下のデータを送信してファームウェアアップデートが完了します。

```
<0x20>Loading...<0x20><0x0D><0x0A>  
<0x20>Loading<0x20>Done<0x0D><0x0A>
```

ファームウェアアップデートが完了することで、PC-WiFi-01 は AT コマンドを受信する準備が整います。

5.6 コマンドリファレンス

PC-WiFi-01 で実行可能なコマンドとその詳細に関しましては、弊社 HP より「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」をダウンロードし、参照してください。本節では、各サンプルプログラムで使用している AT コマンドを抜粋し、以下に紹介します。

※赤文字で書かれている部分に関しては、技術基準適合証明に関わる部分です。PC-WiFi-01,WM-RP-xx は「2.4GHz、1-13CH、送信出力電力 10dBm」にて技術基準適合証明を取得しております。これ以外の動作を行った場合、弊社は一切責任を負いませんのでご了承ください。

5.6.1 Band コマンド

コマンド説明	
説明	使用周波数帯域の設定を行います。
コマンド	at+rsi_band
使用方法	at+rsi_band=band_val<0x0D><0x0A>
パラメータ説明	band_val [※] 0 : 2.4GHz 帯
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A>

※「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」の「2.3.1 Band」では「5GHz」の設定の記載がありますが、「5GHz」は使用出来ません。

5.6.2 Init コマンド

コマンド説明	
説明	at+rsi_band コマンド送信後に要求されるコマンドです。
コマンド	at+rsi_init
使用方法	at+rsi_init<0x0D><0x0A>
パラメータ説明	-
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A> or ERROR<0xF2><0x0D><0x0A>

5.6.3 Scan コマンド

コマンド説明	
説明	指定したチャンネルを走査します。
コマンド	at+rsi_scan
使用方法	at+rsi_scan=chan_num,SSID<0x0D><0x0A>
パラメータ説明	<p><i>chan_num</i> 0 ~ 13 : 走査するチャンネル番号を ASCII で指定します。 (0 : 全チャンネル)</p> <p><i>SSID</i> 対象の SSID を ASCII で指定します。(オプションパラメータ)</p>
レスポンス	<p>正常終了時 OK<uSSID1uMode1RSSIval1><uSSID2uMode2RSSIval2>... <0x0D><0x0A></p> <p>異常終了時 ERROR<0xF8><0x0D><0x0A> or ERROR<0xF3><0x0D><0x0A> or ERROR<0xED><0x0D><0x0A></p>

レスポンス詳細	
uSSID	SSID が ASCII で返されます。32 バイト固定で、不足分は <0x00> で補われます。
uMode	セキュリティモードが 1 バイトの Hex データで返されます。 0x00 : オープン 0x01 : WPA 0x02 : WPA2 0x03 : WEP
RSSIval	RSSI の値が 1 バイトの Hex データで返されます。

5.6.4 Set Network Type コマンド

コマンド説明	
説明	ネットワーク接続形態を設定します。
コマンド	at+rsi_network
使用方法	(インフラストラクチャモード時) at+rsi_network=Parameter<0x0D><0x0A> (アドホックモード時) at+rsi_network=Parameter,type,chan_num<0x0D><0x0A>
パラメータ説明	<i>Parameter</i> ネットワークモードを ASCII で指定します。 INFRASTRUCTURE : インフラストラクチャモード IBSS : アドホック (セキュリティなし) モード IBSS_SEC : アドホック (セキュリティ WEP) モード <i>type</i> IBSS の接続形態を ASCII で指定します。 0 : ジョイナー 1 : クリエータ <i>chan_num</i> IBSS を作成する際のチャンネル番号を ASCII で指定します。 type=1 のときのみ有効で、それ以外のときは 0 を設定してください。
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A>

5.6.5 Pre Shared Key コマンド

コマンド説明	
説明	セキュリティ接続を行う際に必要な PSK (Pre Shared Key) の設定を行います。
コマンド	at+rsi_psk
使用方法	at+rsi_psk=Pre Shared Key<0x0D><0x0A>
パラメータ説明	<i>Pre Shared Key</i> PSK を ASCII で指定します (最大 31 文字)。 ただし、WEP 時は 16 進数での指定となるため、WEP-64bit は 10 桁、WEP-128bit は 26 桁の指定となります。
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A>

5.6.6 Authentication Mode コマンド

コマンド説明	
説明	認証モードの設定を行います。
コマンド	at+rsi_authmode
使用方法	at+rsi_authmode=value<0x0D><0x0A>
パラメータ説明	value 設定値を ASCII で指定します。 0 : Open Key Authentication 1 : Shared Key Authentication
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A>

5.6.7 Join コマンド

コマンド説明	
説明	SSID への接続を行います。
コマンド	at+rsi_join
使用方法	at+rsi_join=SSID name,TxRate,TxPower<0x0D><0x0A>
パラメータ説明	SSID name 接続する SSID を ASCII で指定します (最大 32 文字)。 TxRate 「Table 5.6-1 送信レート表」参照 TxPower [※] 送信時の電力を ASCII で指定します。 1 : Medium Power(10dBm)
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xFD><0x0D><0x0A> or ERROR<0xF8><0x0D><0x0A> or ERROR<0xF4><0x0D><0x0A> or ERROR<0xF1><0x0D><0x0A> or ERROR<0xEE><0x0D><0x0A> or ERROR<0xED><0x0D><0x0A> or ERROR<0xEB><0x0D><0x0A>

※ 「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」の「2.3.12 Join」では「Low power (7dBm)」、
「High power (16 or 17dBm)」の設定の記載がありますが、これらは使用出来ません。

通信レート(Mbps)	802.11 b/g/n	TxRate パラメータ
Auto-rate	-	0
1	b	1
2	b	2
5.5	b	3
11	b	4
6	g	5
9	g	6
12	g	7
18	g	8
24	g	9
36	g	10
48	g	11
54	g	12
MCS0(6.5)	n	13
MCS1(13)	n	14
MCS2(19.5)	n	15
MCS3(26)	n	16
MCS4(39)	n	17
MCS5(52)	n	18
MCS6(58.5)	n	19
MCS7(65)	n	20

Table 5.6-1 送信レート表

5.6.8 Set IP Parameters コマンド

コマンド説明	
説明	IP アドレス、サブネットマスク、デフォルトゲートウェイの設定を行います。
コマンド	at+rsi_ipconf
使用方法	at+rsi_ipconf=DHCP_MODE,IP address,SUBNET,GATEWAY <0x0D><0x0A>
パラメータ説明	<p>DHCP_MODE DHCP の設定を ASCII で指定します。 0 : Manual 1 : DHCP 2 : Auto-IP</p> <p>IP Address IP アドレスを ASCII で指定します。DHCP 時は「0」としてください。</p> <p>SUBNET サブネットマスクを ASCII で指定します。DHCP 時は「0」としてください。</p> <p>GATEWAY ゲートウェイを ASCII で指定します。DHCP 時は「0」としてください。</p>
レスポンス	<p>正常終了時 OK<MAC_Address><IP_Address><SUBNET><Gateway><0x0D><0x0A> A> MAC_Address 6 バイトの MAC アドレスが Hex データで返されます。 IP_Address 4 バイトの IP アドレスが返されます。 SUBNET 4 バイトのサブネットマスクが返されます。 Gateway 4 バイトのゲートウェイが返されます。</p> <p>異常終了時 ERROR<0xFC><0x0D><0x0A> or ERROR<0xFB><0x0D><0x0A> or ERROR<0xF9><0x0D><0x0A> or ERROR<0xF8><0x0D><0x0A> or ERROR<0xF6><0x0D><0x0A> or ERROR<0xF0><0x0D><0x0A> or ERROR<0x7E><0x0D><0x0A> or ERROR<0x7D><0x0D><0x0A></p>

5.6.9 Open a Listening TCP Server コマンド

コマンド説明	
説明	TCP/IP Server 接続を行います
コマンド	at+rsi_ltcp
使用方法	at+rsi_ltcp=port<0x0D><0x0A>
パラメータ説明	port 自分のポート番号を ASCII で指定します。
レスポンス	正常終了時 OK<sock_handle><0x0D><0x0A> sock_handle オープンしたソケット番号が 1 バイトの Hex データで返されます。 異常終了時 ERROR<0xF8><0x0D><0x0A> or ERROR<0x81><0x0D><0x0A>

5.6.10 Open a Listening UDP Socket コマンド

コマンド説明	
説明	UDP Server 接続を行います
コマンド	at+rsi_ludp
使用方法	at+rsi_ludp=lport<0x0D><0x0A>
パラメータ説明	lport 自分のポート番号を ASCII で指定します。
レスポンス	正常終了時 OK<sock_handle><0x0D><0x0A> sock_handle オープンしたソケット番号が 1 バイトの Hex データで返されます。 異常終了時 ERROR<0xF8><0x0D><0x0A> or ERROR<0x81><0x0D><0x0A>

5.6.11 Send data to a Socket コマンド

コマンド説明	
説明	オープンしたソケットからデータを送信します。
コマンド	at+rsi_snd
使用方法	at+rsi_snd=hn,sz,Dip,Dport,stream<0x0D><0x0A>
パラメータ説明	<p><i>hn</i> 送信に用いるソケット番号を ASCII で指定します。</p> <p><i>sz</i> 送信サイズを ASCII で指定します (最大 1400 バイト)。</p> <p><i>Dip</i> 送信先の IP アドレスを ASCII で指定します。(TCP/IP ソケット通信の場合は 0 を指定します)</p> <p><i>Dport</i> 送信先のポート番号を ASCII で指定します。(TCP/IP ソケット通信の場合は 0 を指定します)</p> <p><i>stream</i>* 送信データです。</p>
レスポンス	<p>正常終了時 OK<sz><0x0D><0x0A> sz 実際に送信したサイズが 2 バイトの Hex データで返されます。</p> <p>異常終了時 ERROR<0xFE><0x0D><0x0A> or ERROR<0xF8><0x0D><0x0A> or ERROR<0xF7><0x0D><0x0A> or ERROR<0xE0><0x0D><0x0A> or ERROR<0x1F><0x0D><0x0A></p>

※ 送信したいバイナリデータ上に<0x0D><0x0A>が存在する場合、<0x0D><0x0A>をコマンドの終端として使用されている為、そのままでは送信できません。

<0x0D><0x0A>を送信する場合は、送信データ stream を以下の青背景のデータを赤背景のデータへと変換してから送信してください。

- 例 1) <0x41><0x42><0x43><0x0D><0x0A>を送信する場合
 <0x0D><0x0A>の部分を<0xDB><0xDC>と置き換えて、
 <0x41><0x42><0x43><0xDB><0xDC>として送信する。
- 例 2) <0x41><0x42><0x43><0x0D><0x0A><0x31><0x32>を送信する場合
 <0x0D><0x0A>の部分を<0xDB><0xDC>と置き換えて、
 <0x41><0x42><0x43><0xDB><0xDC><0x31><0x32>として送信する。
- 例 3) <0x41><0x42><0x43><0xDB><0x31><0x32>を送信する場合
 <0xDB>の部分を<0xDB><0xDD>として、
 <0x41><0x42><0x43><0xDB><0xDD><0x31><0x32>として送信する。
- 例 4) <0x41><0x42><0x43><0xDB><0xDC><0x31><0x32>を送信する場合
 <0xDB><0xDC>の部分を<0xDB><0xDD><0xDC>として、
 <0x41><0x42><0x43><0xDB><0xDD><0xDC><0x31><0x32>として送信する。

- 例 5) <0x41><0x42><0x43><0x0D><0x0A><0xDB><0x31><0x32>を送信する場合
 <0x0D><0x0A><0xDB>の部分を<0xDB><0xDC><0xDB><0xDD>として、
 <0x41><0x42><0x43><0xDB><0xDC><0xDB><0xDD><0x31><0x32>として送信する。
- 例 6) <0x41><0x42><0x43><0x0D><0x0A><0xDB><0xDC><0x31><0x32>をする場合
 <0x0D><0x0A><0xDB><0xDC>の部分を<0xDB><0xDC><0xDB><0xDD><0xDC>として、
 <0x41><0x42><0x43><0xDB><0xDC><0xDB><0xDD><0xDC><0x31><0x32>として送信する。

5.6.12 Receive data on a Socket

コマンド説明	
説明	オープンしたソケットからの受信を行います。
コマンド	_※
使用方法	_※
パラメータ説明	-
レスポンス	AT+RSI_READ<hn><sz><Sip><Sport><stream><0x0D><0x0A> hn 受信したソケットの番号が 1 バイトの Hex データで与えられます。 sz 受信したサイズが 2 バイトの Hex データで与えられます。 Sip 受信元の IP アドレスが 4 バイトの Hex データで与えられます。 Sport 受信元のポート番号が 2 バイトの Hex データで与えられます。 stream 受信したデータのストリームが ASCKII で与えられます。

- ※ 本項目はコマンドではありません。
 PC-WiFi-01 が接続先からデータを受信した場合、PC-WiFi-01 が自動的にホスト CPU へとレスポンスデータを送信します。

5.6.13 Close a Socket コマンド

コマンド説明	
説明	オープンしたソケットのクローズを行います。
コマンド	at+rsi_cls [※]
使用方法	at+rsi_cls=hn<0x0D><0x0A> [※]
パラメータ説明	hn クローズするソケット番号を ASCII で指定します。
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A> or ERROR<0xFA><0x0D><0x0A> or ERROR<0x40><0x0D><0x0A>

- ※ 接続相手側が切断を行った場合、PC-WiFi-01 は自動的にホスト CPU へと「AT+RSI_CLOSE<sock_handle><0x0D><0x0A>」を送信します。
(sock_handle : クローズされたソケット番号を指す 1 バイトの Hex データ)

5.6.14 Disassociate コマンド

コマンド説明	
説明	接続しているアクセスポイントからの切断を行います。
コマンド	at+rsi_disassoc
使用方法	at+rsi_disassoc<0x0D><0x0A>
パラメータ説明	-
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A>

- ※ アドホックモードでは非サポートのコマンドです。

5.6.15 Soft Reset コマンド

コマンド説明	
説明	Soft Reset を行います。
コマンド	at+rsi_reset
使用方法	at+rsi_reset<0x0D><0x0A>
パラメータ説明	-
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A>

5.6.16 Query Firmware Version コマンド

コマンド説明	
説明	モジュールのファームウェアバージョンを取得します
コマンド	at+rsi_fwversion
使用方法	at+rsi_fwversion?<0x0D><0x0A>
パラメータ説明	-
レスポンス	正常終了時 OK<Major>.<Minor1>.<Minor2><0x0D><0x0A> Major メジャーバージョン情報を ASCII で返されます。 Minor1, Minor2 マイナーバージョン情報を ASCII で返されます。 異常終了時 ERROR<0xF8><0x0D><0x0A>

5.6.17 Power Mode コマンド

コマンド説明	
説明	モジュールのパワーモードを選択します
コマンド	at+rsi_pwmode
使用方法	at+rsi_pwmode= <i>power_mode</i> <0x0D><0x0A>
パラメータ説明	<i>power_mode</i> 0 : モード 0 1 : モード 1 2 : モード 2
レスポンス	正常終了時 OK<0x0D><0x0A> 異常終了時 ERROR<0xF8><0x0D><0x0A>

モード詳細	
モード 0	デフォルトのパワーモードです。省電力機能はありません。
モード 1	ベースバンド、RF、コアコントロールブロックをスリープさせます。ホストは、モジュールから送られる“SLEEP”の受信と、“ACK”の送信によって、スリープをコントロールします。
モード 2	ベースバンドと RF をスリープさせます。ホストと通信するコアコントロールブロックは常に機能するため、モジュールはいつでもホストからコマンドを受け取ることが可能で、ホストとの“SLEEP”と“ACK”のやり取りはありません。

- ※ モード 1、2 は、WiFi 接続前と WiFi 接続後ではそれぞれ動作が異なります。本サンプルプログラムでは WiFi 接続前にモード 1 に変更していますので、この場合の動作について Fig 5.6-1 ~ Fig 5.6-3 に簡単な制御フローを示します。これらの詳細およびモード 1 の WiFi 接続後やその他のモードに関しては、「RS9110-N-11-22_24_26_28_Software_PRM-v2.54a.pdf」の「2.4 Power Modes and commands」を参照してください。

- Power Mode1 WiFi 接続前 制御フロー (Power Mode 開始)

ホスト CPU が pwmode コマンドを送信すると、モジュールから“OK¥r¥n”が送信され、PowerMode1 が開始されます。
その後モジュールから“SLEEP¥r¥n”を受信します。

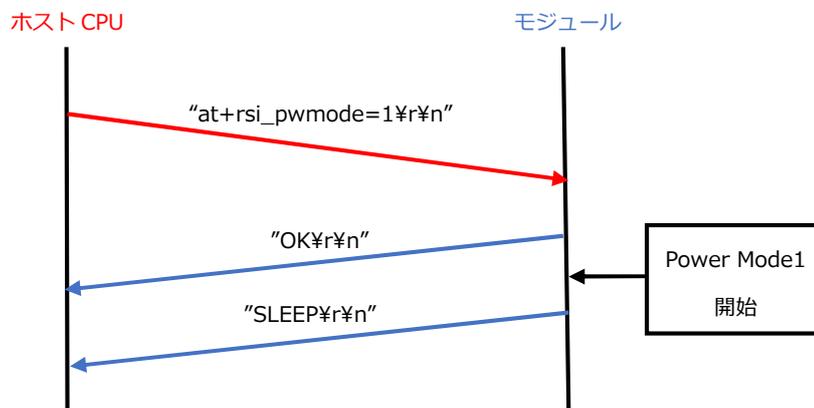


Fig 5.6-1 Power Mode1 WiFi 接続前 制御フロー (Power Mode 開始)

- Power Mode1 WiFi 接続前 制御フロー (ACK 応答)

スリープさせる場合、ホスト CPU は“SLEEP¥r¥n”を受信した後モジュールに“ACK¥r¥n”を送信します。
スリープ時間終了後、モジュールから“SLEEP¥r¥n”が送信されます。

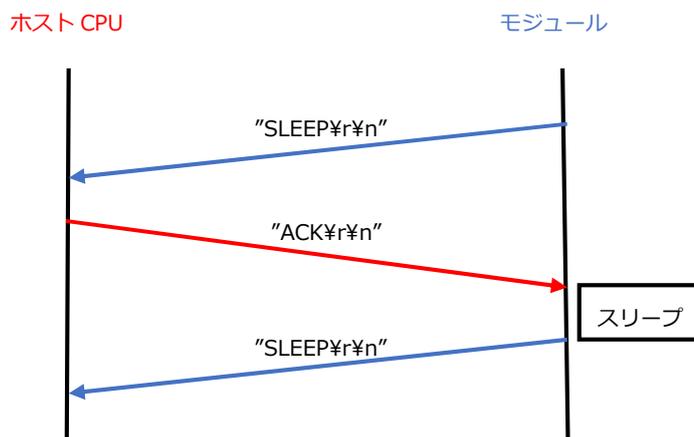


Fig 5.6-2 Power Mode1 WiFi 接続前 制御フロー (ACK 応答)

- Power Mode1 WiFi 接続前 制御フロー (コマンド送信)

WiFi 関連のコマンドを送信する場合、ホスト CPU は“SLEEP¥r¥n”を受信した後に目的のコマンドを送信します (図中では band コマンド)。コマンド実行後、モジュールから“SLEEP¥r¥n”が送信されます。

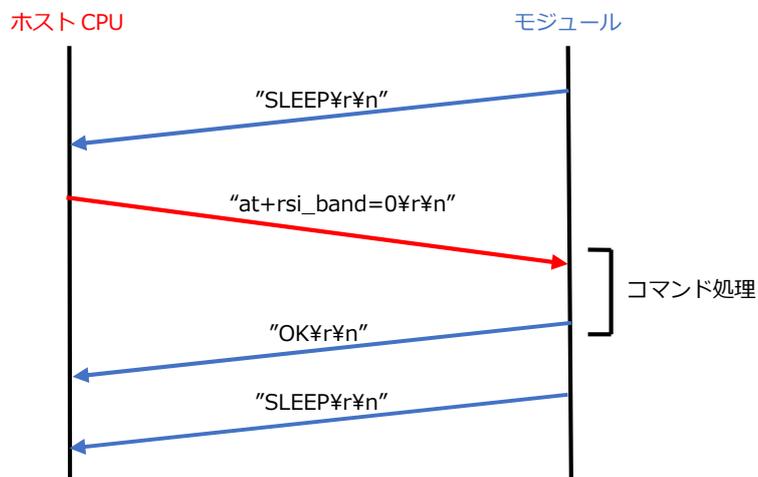


Fig 5.6-3 Power Mode1 WiFi 接続前 制御フロー (コマンド送信)

改定履歴

版数	日付	改定内容
1 版	2011/08/10	新規作成 (サンプルプログラム Ver1.0 以降に対応)
1.1 版	2017/09/04	サンプルプログラム Ver1.1 以降に対応 モジュール起動設定 追加 (全章) 「3.2 動作説明 (TCP/IP 通信)」 更新 (3 章) 「4.2 動作説明 (UDP 通信)」 更新 (4 章) 「5. PC-WIFI-01 制御方法」 更新 (5 章) 誤字修正 (全章)

ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有します。
- ・本文書に記載されている内容およびサンプルプログラムについての技術サポートは一切受け付けておりません。
- ・本サンプルプログラムに関して、株式会社ルネサスエレクトロニクスへのお問い合わせはご遠慮ください。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社および株式会社ルネサスエレクトロニクスでは一切責任を負いませんのでご了承ください。
- ・本文書の内容については、万全を期して作成いたしました。が、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡ください。
- ・本文書の内容は、将来予告なしに変更されることがあります。

商標について

- ・SH7211 は、株式会社ルネサスエレクトロニクスの登録商標、商標または商品名称です
- ・SuperH は、株式会社ルネサスエレクトロニクスの登録商標、商標または商品名称です。

- ・その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市東区積志町 8 3 4
<http://www.apnet.co.jp>
E-MAIL : query@apnet.co.jp